EXHIBIT 037

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
10. A method for buffering data in an integrated circuit having a plurality of processing modules being connected with an interconnect through interface units, wherein a first processing module communicates to a second processing module using transactions, the method comprising the acts of:	Without conceding that the preamble of claim 10 of the '800 Patent is limiting, the Motorola Edge+ Gen 2 (hereinafter, the "Motorola product") performs a method for buffering data in an integrated circuit having a plurality of processing modules being connected with an interconnect through interface units, wherein a first processing module communicates to a second processing module using transactions), either literally or under the doctrine of equivalents. The Motorola product includes an integrated circuit. For example, the Motorola product includes the Qualcomm Snapdragon 8 Gen 1 Mobile Platform system on chip (hereinafter, the "Snapdragon SoC").

[1] The Motorola product is charted as a representative product made used, sold, offered for sale, and/or imported by Motorola. The citations to evidence contained herein are illustrative and should not be understood to be limiting. The right is expressly reserved to rely upon additional or different evidence, or to rely on additional citations to the evidence cited already cited herein.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **Motorola Edge+ Gen 2** **Featuring a Snapdragon 8 Gen 1 Mobile Platform** The Motorola edge+ was born for 5G speed. This state-of-the-art smartphone gives you up to 2 full days of power, lightning-fast speed, and pro-quality features for doing more of what you love. Leave lag time behind with a massive 256 GB+ memory and blazing-fast premium Snapdragon mobile platform. Enjoy days of entertainment on a beautiful display that wraps around the edges and has superior stereo-quality sound. Get the best of Android OS without the extra baggage. Learn more https://www.qualcomm.com/snapdragon/device-finder/motorola-edge--gen-2 The Snapdragon SoC comprises a plurality of processing modules, for example Qualcomm Adreno GPU; Qualcomm Kryo CPU; Qualcomm Hexagon Processor; and Platform Security Foundations, Trusted Execution Environment & Services, Secure Processing Unit (SPU):

4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]

Snapdragon
8 mobile platform
Gen 1

SPECIFICATIONS & FEATURES

Artificial Intelligence

Qualcomm® Adreno™ GPU

Qualcomm® Kryo™ CPU

Qualcomm® Hexagon™ Processor
- Fused AI Accelerator
  - Hexagon Tensor Accelerator
  - Hexagon Vector eXtensions
  - Hexagon Scalar Accelerator
- Support for mix precision(INT8+INT16)
- Support for all precisions (INT8, INT16, FP16)

Qualcomm® Sensing Hub

5G Modem-RF System

Snapdragon X65 5G Modem-RF System
- 5G mmWave and sub-6 GHz, standalone (SA) and non-standalone (NSA) modes, FDD, TDD
- Dynamic Spectrum Sharing
- mmWave: 1000 MHz bandwidth, 8 carriers, 2x2 MIMO
- Sub-6 GHz: 300 MHz bandwidth, 4x4 MIMO
- Qualcomm® 5G PowerSave 2.0
- Qualcomm® Smart Transmit™ 2.0 technology
- Qualcomm® Wideband Envelope Tracking
- Qualcomm® AI-Enhanced Signal Boost
- Global 5G multi-SIM

Downlink: Up to 10 Gbps

Multimode support: 5G NR, LTE including CBRS, WCDMA, HSPA, TD-SCDMA, CDMA 1x, EV-DO, GSM/EDGE

Camera

Qualcomm Spectra™ Image Signal Processor
- Triple 18-bit ISPs
- Up to 3.2 Gigapixels per Second computer vision ISP (CV-ISP)
- Up to 36 MP triple camera @ 30 FPS with Zero Shutter Lag
- Up to 64+36 MP dual camera @ 30 FPS with Zero Shutter Lag
- Up to 108 MP single camera @ 30 FPS with Zero Shutter Lag
- Up to 200 Megapixel Photo Capture

Rec. 2020 color gamut photo and video capture

Up to 10-bit color depth photo and video capture

8K HDR Video Capture + 64 MP Photo Capture

10-bit HEIF: HEIC photo capture, HEVC video capture

Video Capture Formats: HDR10+, HDR10, HLG, Dolby Vision

8K HDR Video Capture @ 30 FPS

4K Video Capture @ 120 FPS

Slow-mo video capture at 720p @ 960 FPS

Bokeh Engine for Video Capture

Video super resolution

Multi-frame Noise Reduction (MFNR)

Locally Motion Compensated Temporal Filtering

Multi-Frame and triple exposure staggered/digital overlap HDR dual-sensor support

AI-based face detection, auto-focus, and auto-exposure

CPU

Kryo CPU
- Up to 3.0 GHz*, with Arm Cortex-X2 technology
- 64-bit Architecture

Visual Subsystem

Adreno GPU
- Vulkan® 1.1 API support
- HDR gaming (10-bit color depth, Rec. 2020 color gamut)
- Physically Based Rendering
- Volumetric Rendering
- Adreno Frame Motion Engine
- API Support: OpenGL® ES 3.2, OpenCL™ 2.0 FP, Vulkan 1.1
- Hardware-accelerated H.265 and VP9 decoder
- HDR Playback Codec support for HDR10+, HDR10, HLG and Dolby Vision

Security

Platform Security Foundations, Trusted Execution Environment & Services, Secure Processing Unit (SPU)

Trust Management Engine

Qualcomm® wireless edge services (WES) and premium security features

Qualcomm® 3D Sonic Sensor and Qualcomm 3D Sonic Max (fingerprint sensor)

Qualcomm® Type-1 Hypervisor

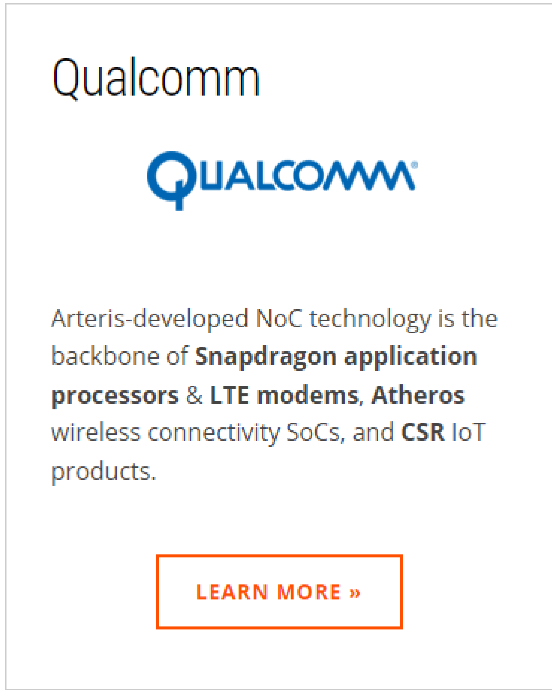
4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Wi-Fi & Bluetooth* Qualcomm® FastConnect™ 6900 System • Wi-Fi Standards: Wi-Fi 6E, Wi-Fi 6 (802.11ax), Wi-Fi 5 (802.11ac), 802.11a/b/g/n • Wi-Fi Spectral Bands: 2.4 GHz, 5 GHz, 6 GHz • Peak speed: 3.6 Gbps • Channel Bandwidth: 20/40/80/160 MHz • 8-stream sounding (for 8x8 MU-MIMO) • MIMO Configuration: 2x2 (2-stream) • MU-MIMO (Uplink & Downlink) • 4K QAM • OFDMA (Uplink & Downlink) • Dual-band simultaneous (2x2 + 2x2) • Wi-Fi Security: WPA3-Enterprise, WPA3- Enhanced Open, WPA3 Easy Connect, WPA3-Personal Integrated Bluetooth • Bluetooth Features: Bluetooth 5.2, LE Audio, Dual Bluetooth antennas • Bluetooth audio: Snapdragon Sound™ Technology with support for Qualcomm® aptX™ Voice, aptX Lossless, aptX Adaptive, and LE audio **Audio** Qualcomm Aqstic™ audio codec (WCD9385) New Qualcomm Aqstic smart speaker amplifier (WSA8835) Total Harmonic Distortion + Noise (THD+N), Playback: -108dB Qualcomm Audio and Voice Communication Suite **Display** On-Device Display Support: • 4K @ 60 Hz • QHD+ @ 144 Hz Maximum External Display Support: up to 4K @ 60 Hz • 10-bit color depth, Rec. 2020 color gamut • HDR10 and HDR10+ Demura and subpixel rendering for OLED Uniformity **Charging** Qualcomm® Quick Charge™ 5 Technology **Location** GPS, Glonass, BeiDou, Galileo, QZSS, NavIC capable Dual Frequency GNSS (L1/L5) Sensor-Assisted Positioning • Urban pedestrian navigation with sidewalk accuracy • Global freeway lane-level vehicle navigation **Memory** Support for LP-DDR5 memory up to 3200 MHz Memory Density: up to 16 GB **General Specifications** Full Suite of Snapdragon Elite Gaming™ features 4 nm Process Technology USB Version 3.1; USB Type-C Support Part Number: SM8450 **snapdragon.com** * Exact speed measured at 2.995 GHz Certain optional features available subject to Carrier and OEM selection for an additional fee. Snapdragon, Qualcomm, Qualcomm Hexagon, Qualcomm 5G PowerSave, Qualcomm Kryo, Qualcomm Smart Transmit, Qualcomm Wideband Envelope, Qualcomm AI-Enhanced Signal Boost, Qualcomm Spectra, Qualcomm Aqstic, Qualcomm 3D Sonic Sensor, Qualcomm Type-1 Hypervisor, and Qualcomm Quick Charge are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm wireless edge services are offered by Qualcomm Technologies Inc. and/or its subsidiaries. Snapdragon, Qualcomm, Hexagon, Snapdragon Elite Gaming, Adreno, FastConnect, Snapdragon Sound, Kryo, Smart Transmit, Qualcomm Spectra, Qualcomm Aqstic, and Quick Charge are trademarks or registered trademarks of Qualcomm Incorporated. aptX is a trademark or registered trademark of Qualcomm Technologies International, Ltd. ©2021 Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved. https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/snapdragon-8-gen-1-mobile-platform-product-brief.pdf The Snapdragon SoC included in the Motorola product utilizes Arteris network on chip interconnect technology, and/or a derivative thereof, (collectively, the "Arteris NoC") as an interconnect to connect the plurality of processing modules through interface units:

4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 https://web.archive.org/web/20210514110614/https://www.arteris.com/customers

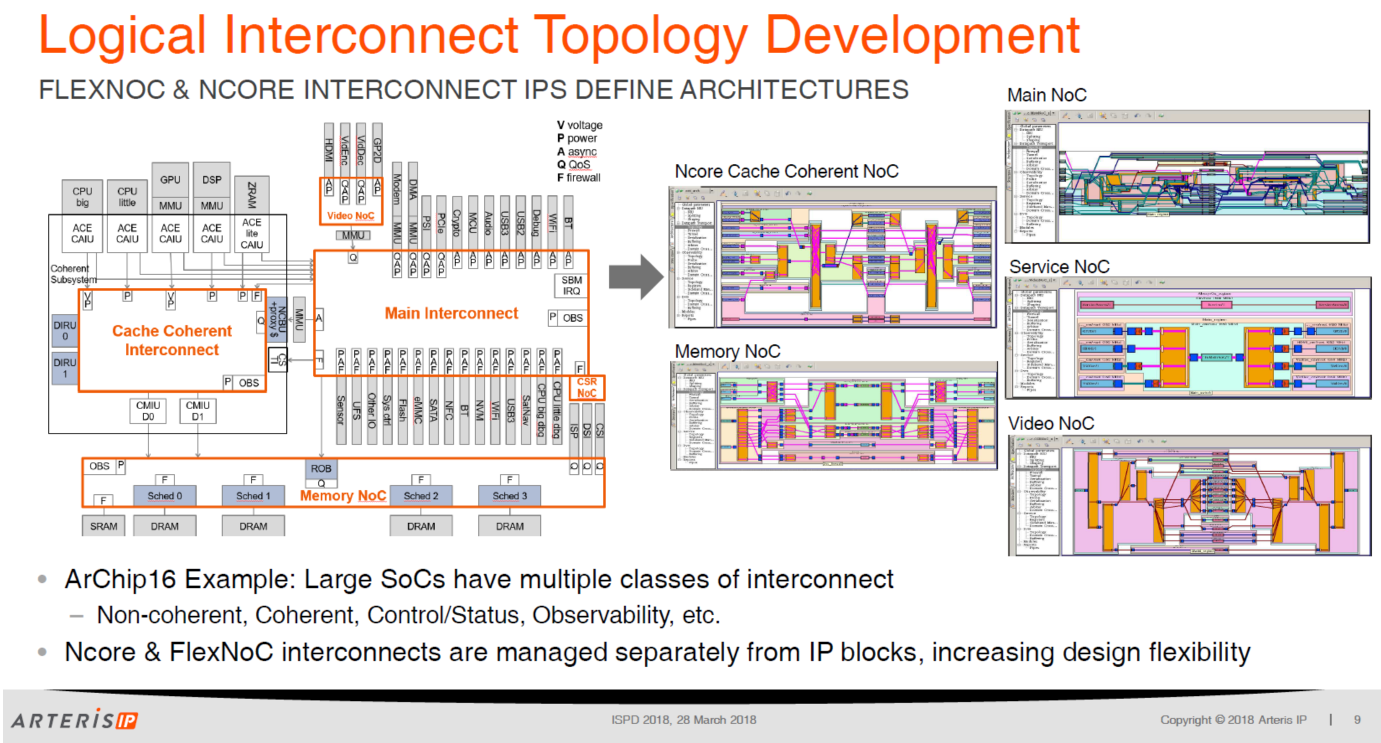
4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Certain Arteris Technology Assets Acquired by **Kurt Shuler**, on October 31, 2013 Arteris to continue to license, support and maintain Arteris FlexNoC® interconnect IP SUNNYVALE, California — October 31, 2013 — Arteris Inc., a leading innovator and supplier of silicon-proven commercial network-on-chip (NoC) interconnect IP solutions, today announced that Qualcomm Technologies, Inc. ("Qualcomm"), a subsidiary of Qualcomm Incorporated, has acquired certain technology assets from Arteris and hired personnel formerly employed by Arteris. *Arteris NoC technology has been and will continue to be a key enabler for creating larger and more complex chips in a shorter amount of time at a lower cost. This acquisition of our technology assets represents a validation of the value of Arteris' Network-on-Chip interconnect IP technology.* *K. Charles Janac, President and CEO,* ***Arteris*** https://www.arteris.com/press-releases/Qualcomm-Arteris-asset-acquisition-2013_oct_31; https://www.fiercewireless.com/tech/qualcomm-acquires-arteris-noc-tech-assets-team A large SoC, such as the Snapdragon SoC included in the Motorola product may include multiple classes of Arteris NoC interconnect:

4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	

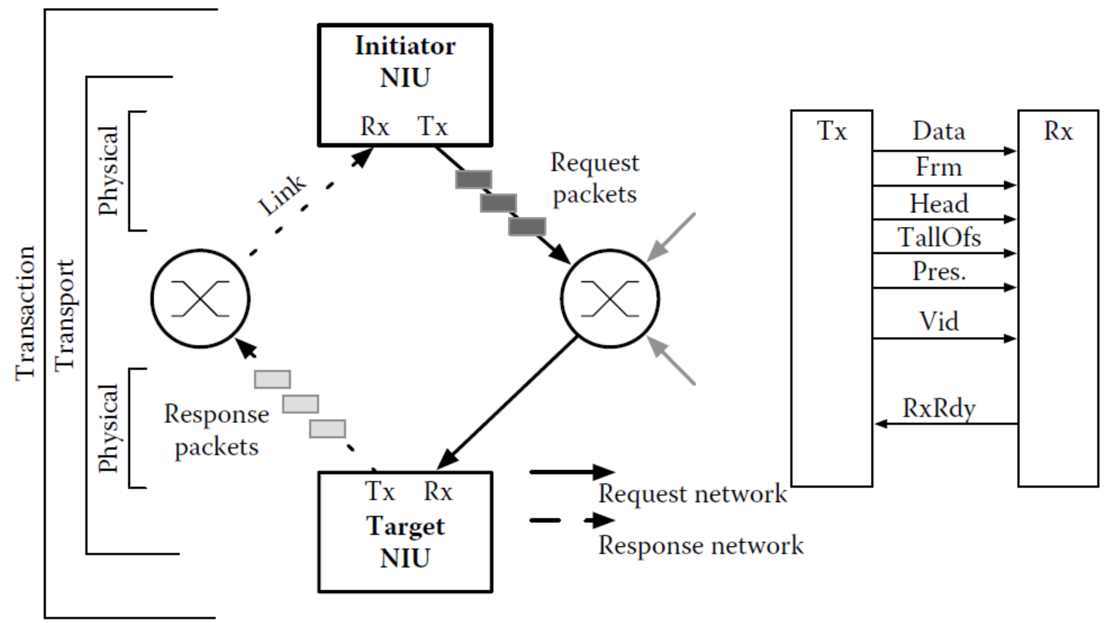
See Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slide 9.

The Arteris NoC is an interconnect connects the plurality of processing modules in the Snapdragon SoC included in the Motorola product through interface units, wherein a first processing module communicates to a second processing module using transactions.

For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets": ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313; *see id* at 308 (explaining that Chapter 11 of this book describes the function of the Arteris NoC: "In this chapter we will present an MPSoC platform […] using Arteris NoC as communication infrastructure.").
configuring the first processing module having a first memory as a	The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product configures the first processing module having a first memory as a master the provides requests, either literally or under the doctrine of equivalents.

4856-4191-2641.3

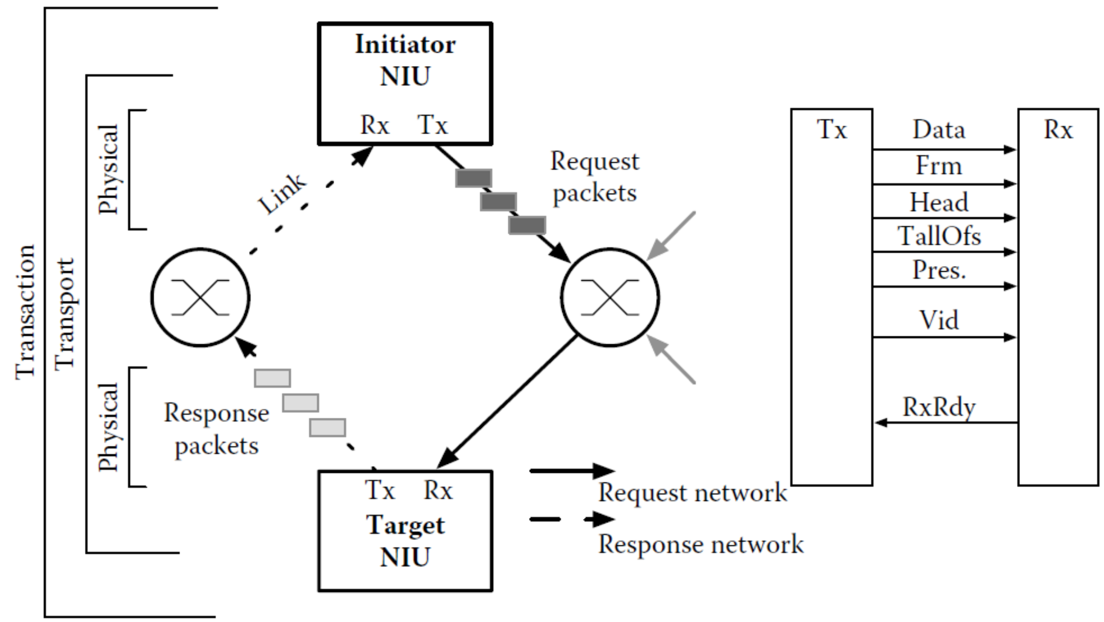
'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
master the provides requests;	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets": ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC":

4856-4191-2641.3

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As a further example, "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC [and] translate[] AHB transactions AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP" and has a "FIFO memory […] inserted in the datapath for AHB write access":

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	11.3.2.1 Initiator NIU Units Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

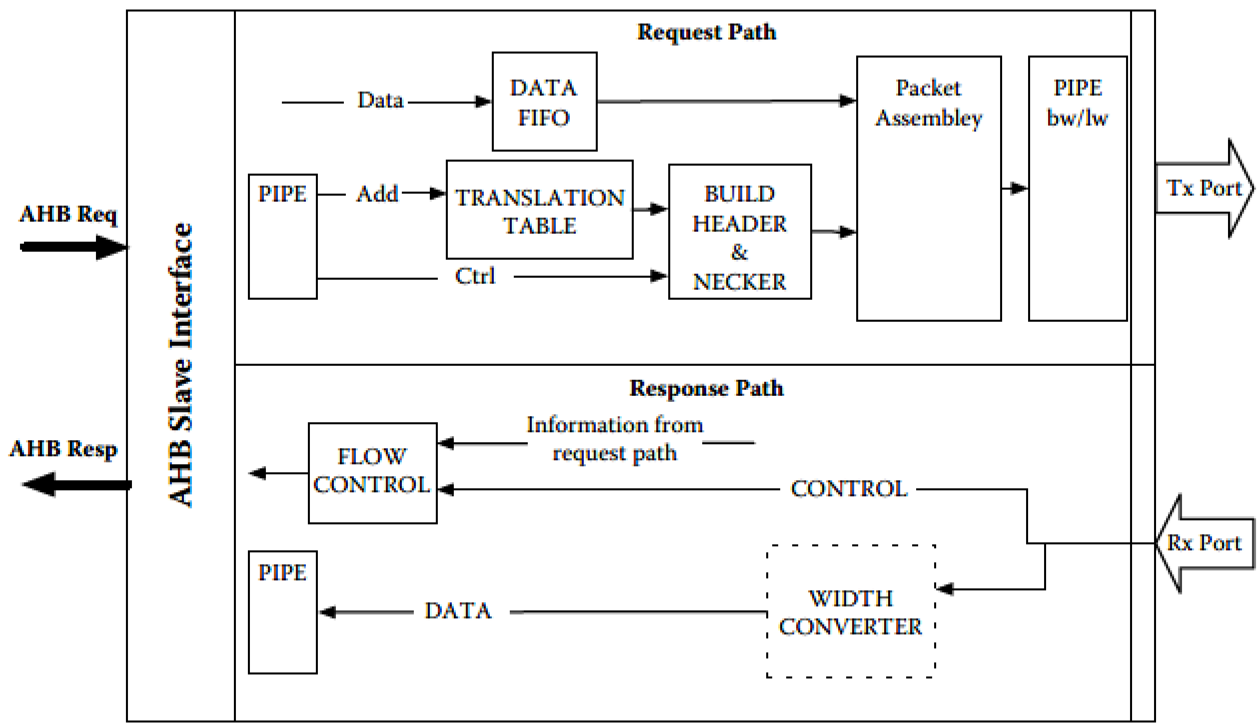
<table>
<tr><th>'800 Patent Claim</th><th>Motorola Product Including Snapdragon System on Chip [1]</th></tr>
<tr><td></td><td>NIU Architecture


FIGURE 11.4
Network interface unit: Initiator architecture.
Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.</td></tr>
<tr><td>configuring the second processing module having a second memory as</td><td>The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product configures the second processing module having a second memory as a slave the provides responses to the requests, either literally or under the doctrine of equivalents.</td></tr>
</table>

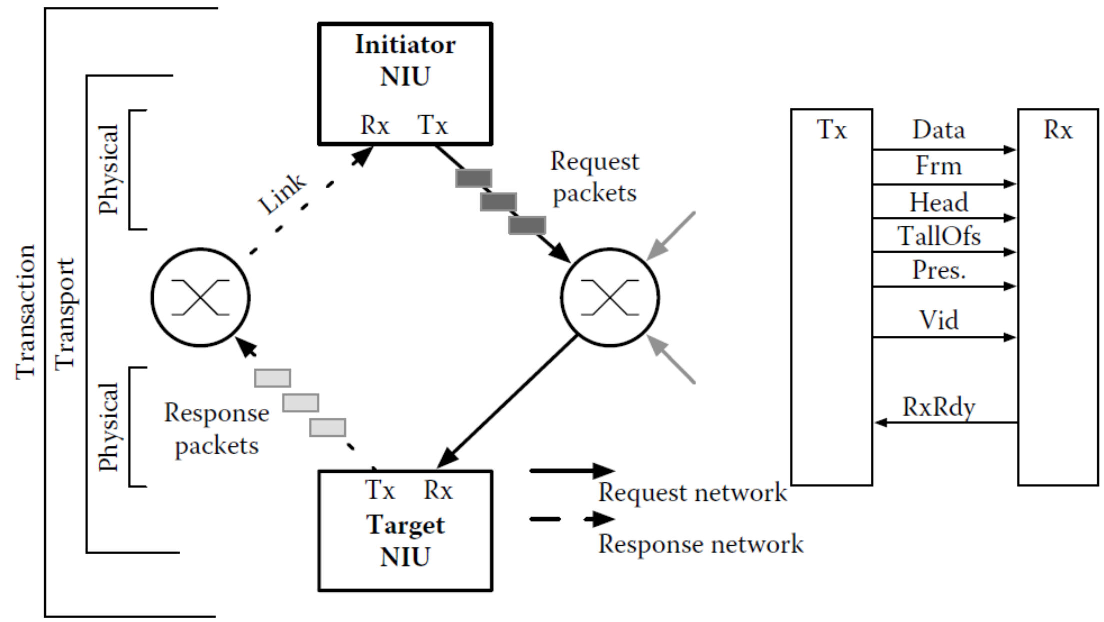
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
a slave the provides responses to the requests;	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets": ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

4856-4191-2641.3

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Target NIUs are "used to connect a slave node to the NoC":

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets" and have a FIFO memory in the datapath:

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	11.3.2.2 Target NIU Units Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

<table>
<tr><th>'800 Patent Claim</th><th>Motorola Product Including Snapdragon System on Chip [1]</th></tr>
<tr><td></td><td>

Target NIU Architecture

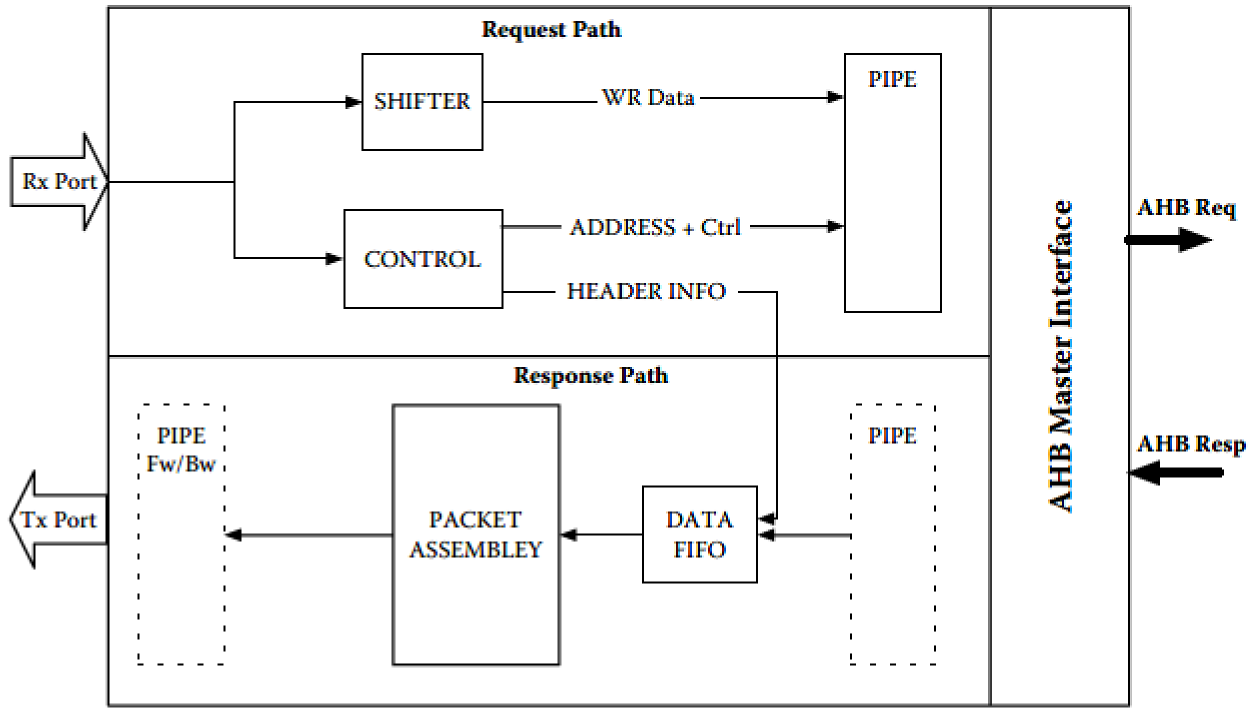


FIGURE 11.5
Network interface unit: Target architecture.

Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.
</td></tr>
<tr><td>connecting the master to a master</td><td>The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product connects the master to a master interface unit of the interface units, either literally or under the doctrine of equivalents.</td></tr>
</table>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
interface unit of the interface units;	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes: ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

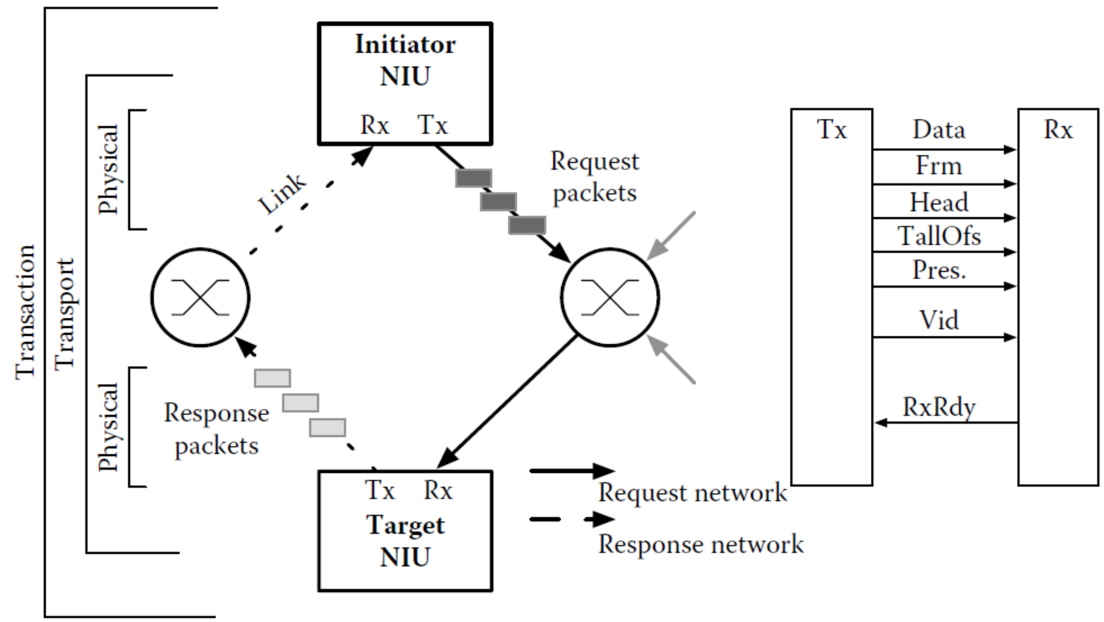
<table>
<tr><th>'800 Patent Claim</th><th>Motorola Product Including Snapdragon System on Chip [1]</th></tr>
<tr><td></td><td>

FIGURE 11.1
NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.

See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.

The Initiator NIUs are "used to connect a master node to the NoC":</td></tr>
</table>

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As a further example, "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC":

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2.1 Initiator NIU Units Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

<table>
<tr><th>'800 Patent Claim</th><th>Motorola Product Including Snapdragon System on Chip [1]</th></tr>
<tr><td></td><td>

NIU Architecture

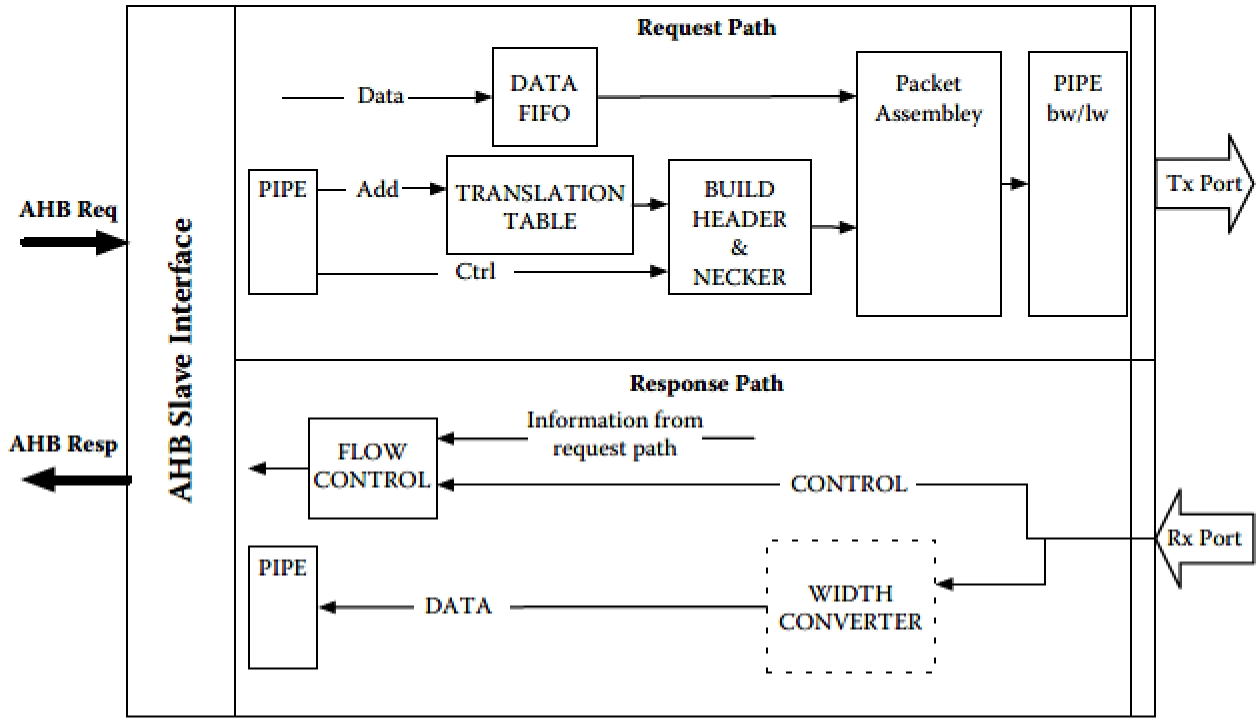


FIGURE 11.4
Network interface unit: Initiator architecture.

Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.
</td></tr>
<tr><td>connecting the master interface unit to the</td><td>The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product connects the master interface unit to the interconnect so that the master interface unit is between the master and the interconnect, either literally or under the doctrine of equivalents.</td></tr>
</table>

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

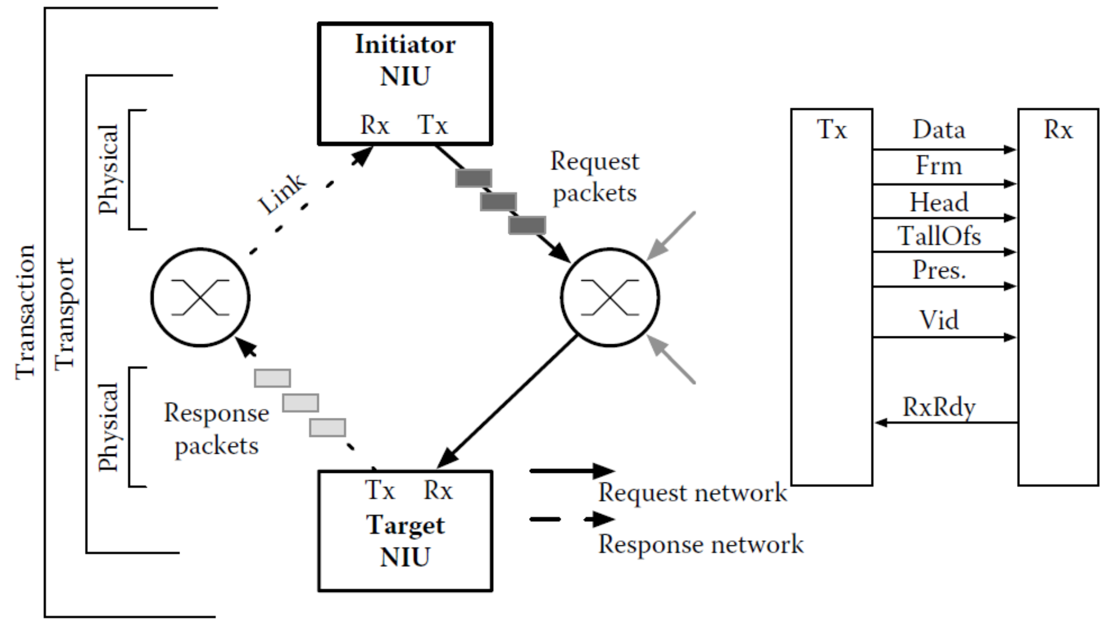
'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
interconnect so that the master interface unit is between the master and the interconnect;	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network: ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC":

4856-4191-2641.3

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As a further example, "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC":

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2.1 Initiator NIU Units Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

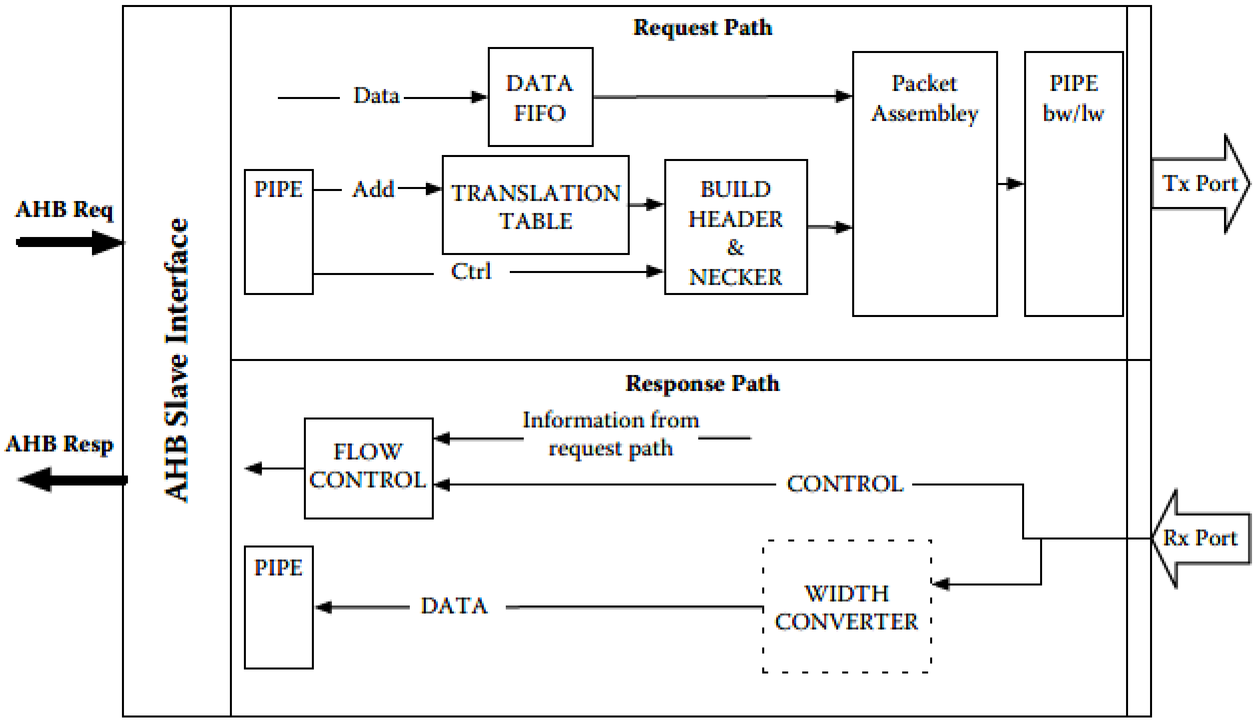
<table>
<tr><th>'800 Patent Claim</th><th>Motorola Product Including Snapdragon System on Chip [1]</th></tr>
<tr><td></td><td>NIU Architecture


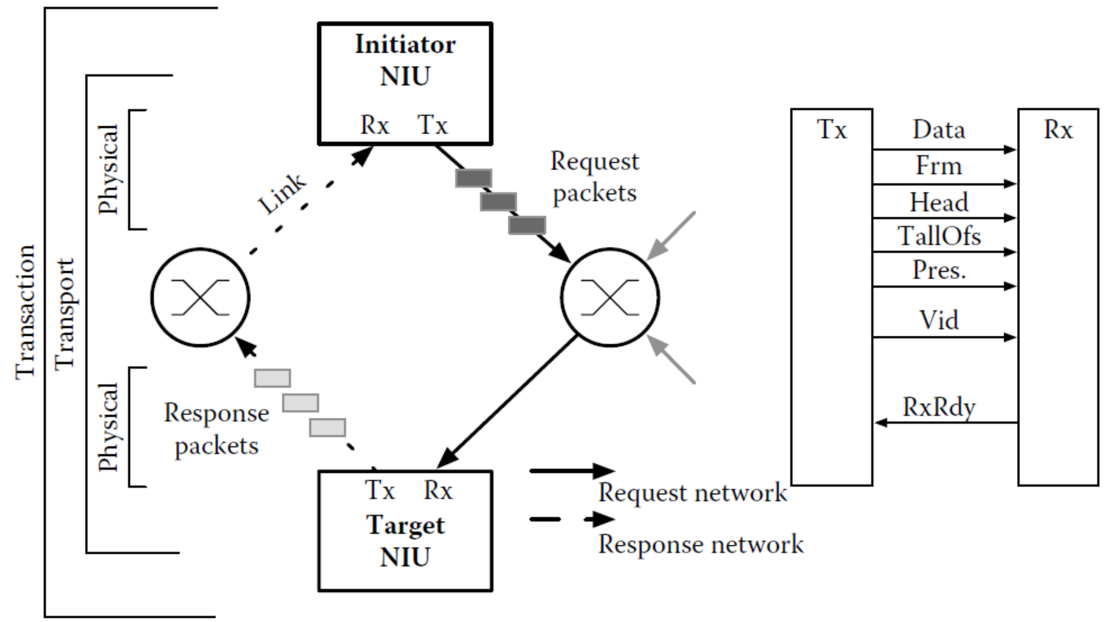
FIGURE 11.4
Network interface unit: Initiator architecture.
Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.</td></tr>
<tr><td>connecting the slave to a slave</td><td>The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product connecting the slave to a slave interface unit of the interface units, either literally or under the doctrine of equivalents.</td></tr>
</table>

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
interface unit of the interface units;	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes: ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

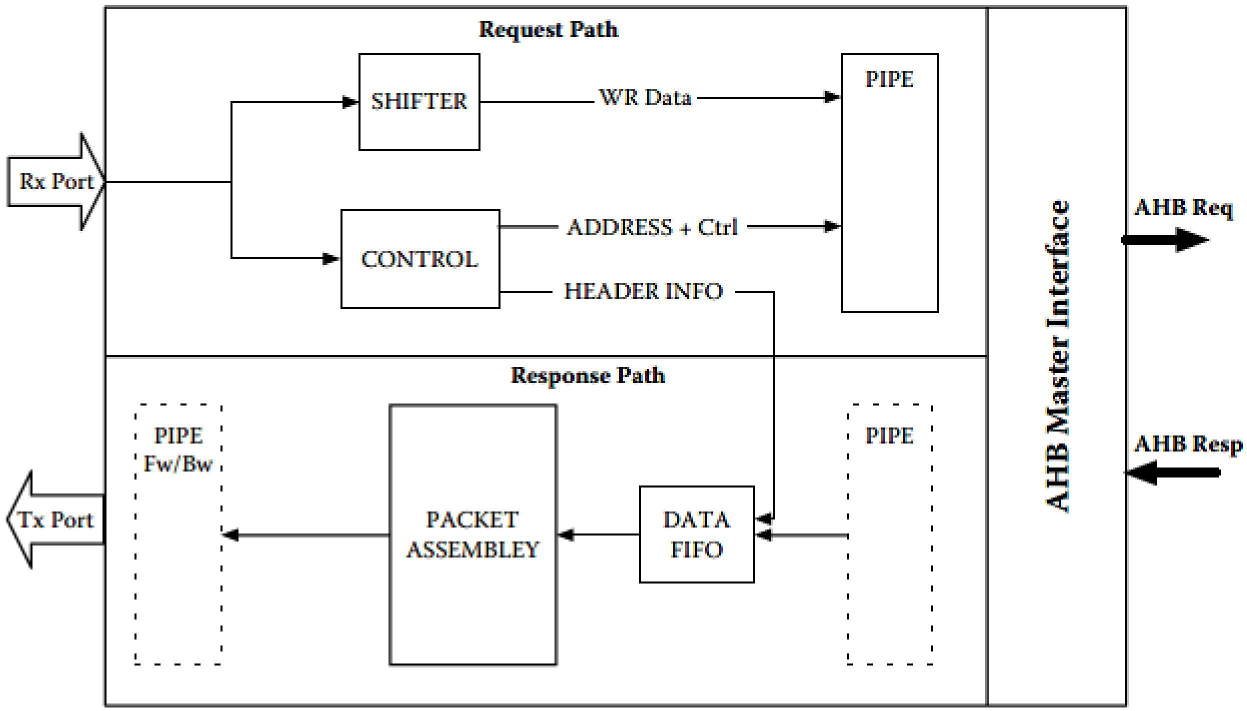
'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Target NIUs are "used to connect a slave node to the NoC":

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets":

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2.2 Target NIU Units Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Target NIU Architecture  **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.

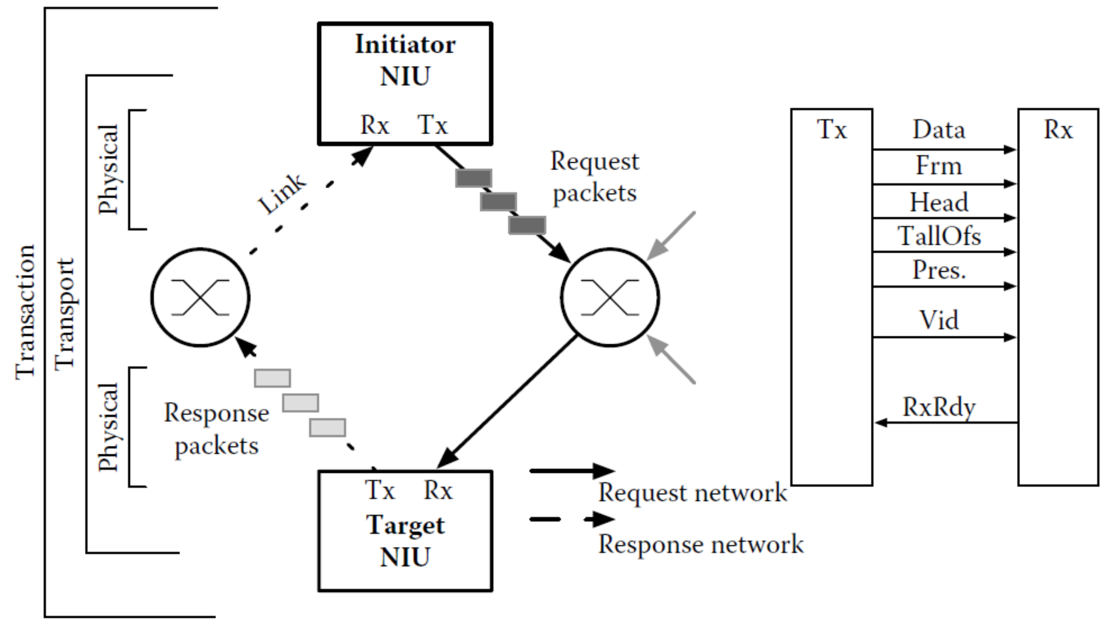
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
connecting the slave interface unit to the interconnect so that the slave interface unit is between the slave and the interconnect;	The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product connects the slave interface unit to the interconnect so that the slave interface unit is between the slave and the interconnect, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network: ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

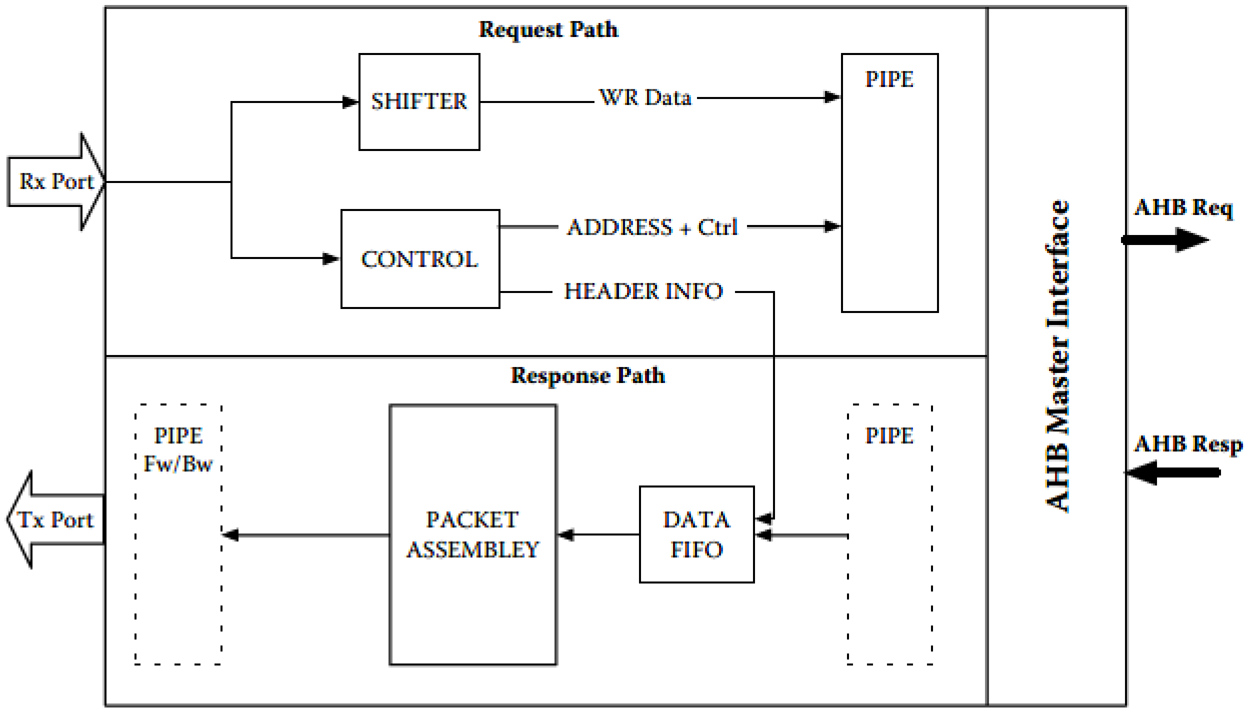
'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Target NIUs are "used to connect a slave node to the NoC":

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets":

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2.2 Target NIU Units Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Target NIU Architecture  **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.
determining by a master	The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product determines by a master determination unit of the master interface unit a first optimal amount of data to be

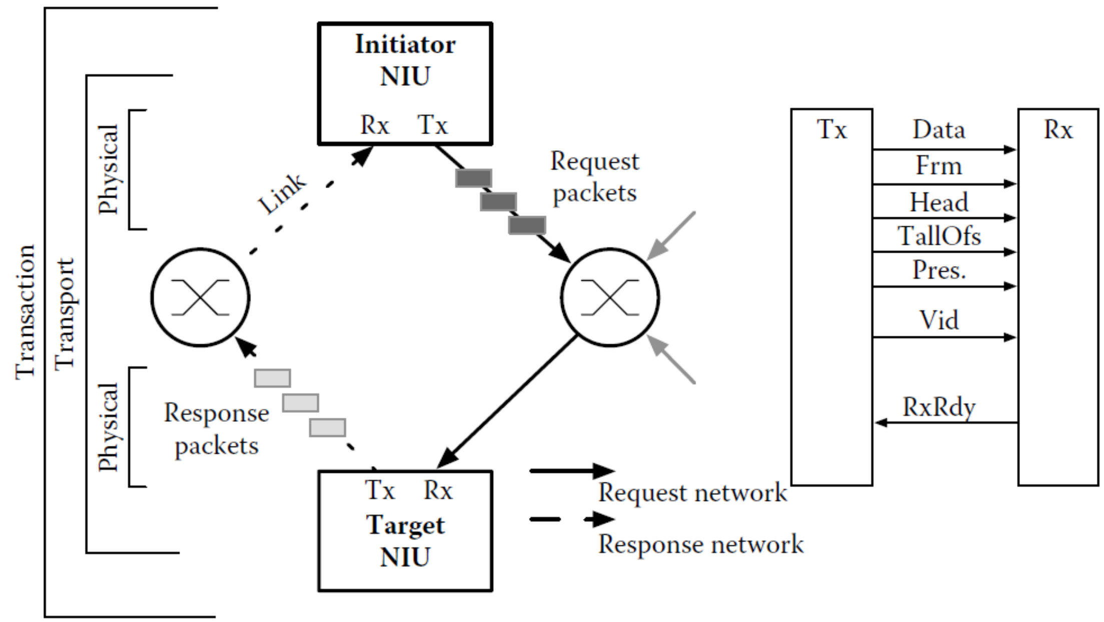
'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
determination unit of the master interface unit a first optimal amount of data to be buffered by a master wrapper of the master interface unit;	buffered by a master wrapper of the master interface unit, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network: ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

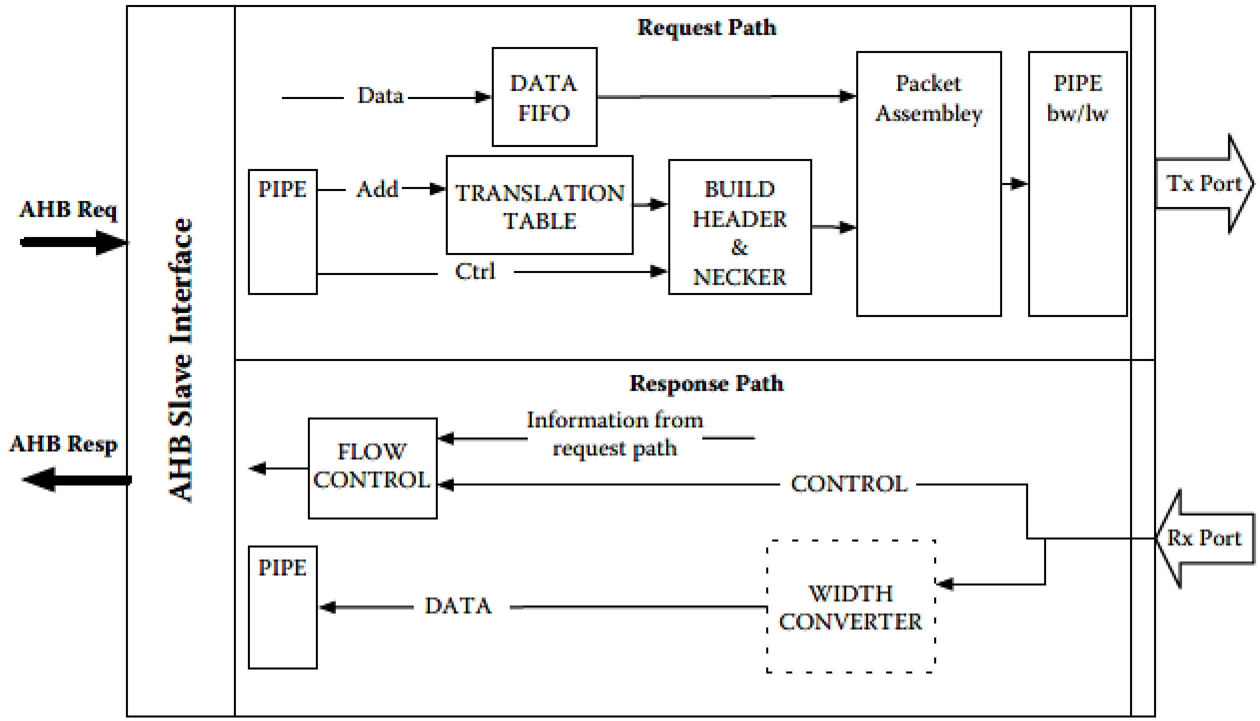
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC":

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. In the Arteris NoC "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC" and includes blocks such as "Data FIFO," "Translation Table," "Build Header & Necker," and "Packet Assembly":

4856-4191-2641.3

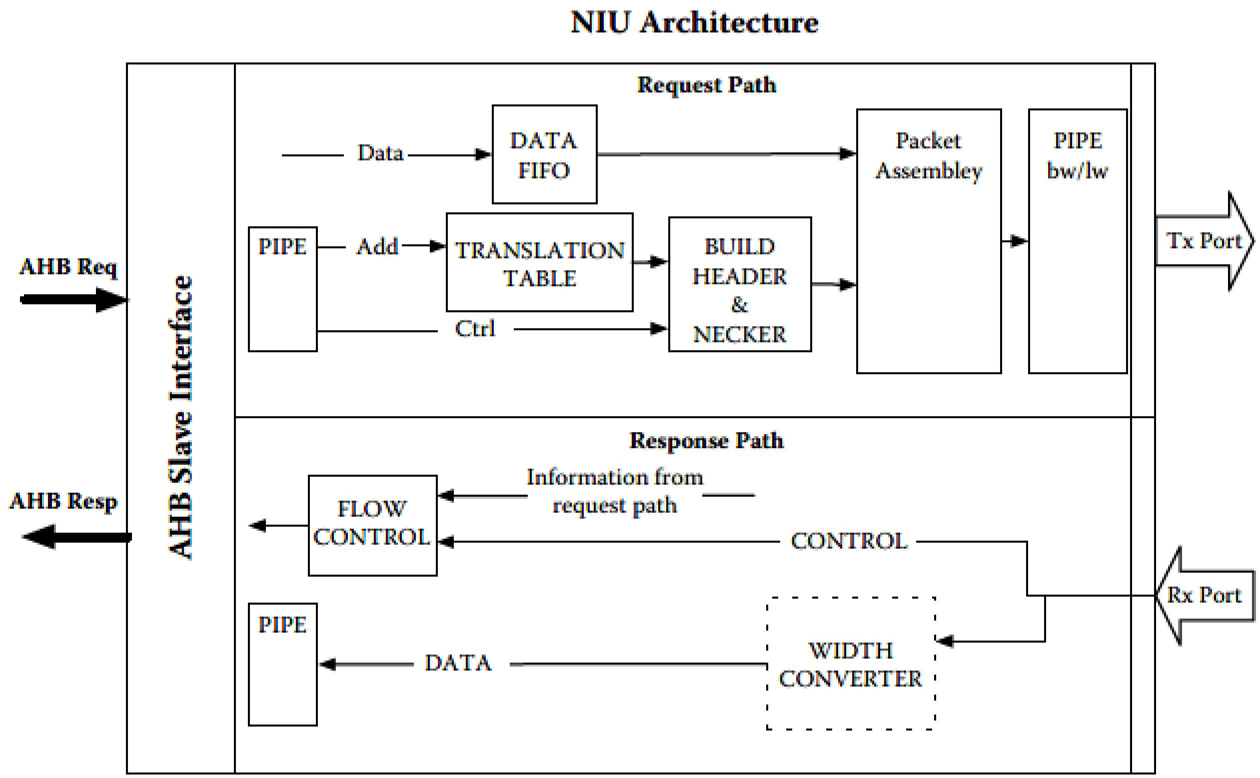
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	NIU Architecture  **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317. In the Initiator NIUs of the Arteris NoC, a "FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	defined by the hardware parameter" which "indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port": ***11.3.2.1 Initiator NIU Units*** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.
determining by a slave determination unit	The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product determines by a slave determination unit of the slave interface unit a second optimal amount of data to be

4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
of the slave interface unit a second optimal amount of data to be buffered by a slave wrapper of the slave interface unit;	buffered by a slave wrapper of the slave interface unit, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network: ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

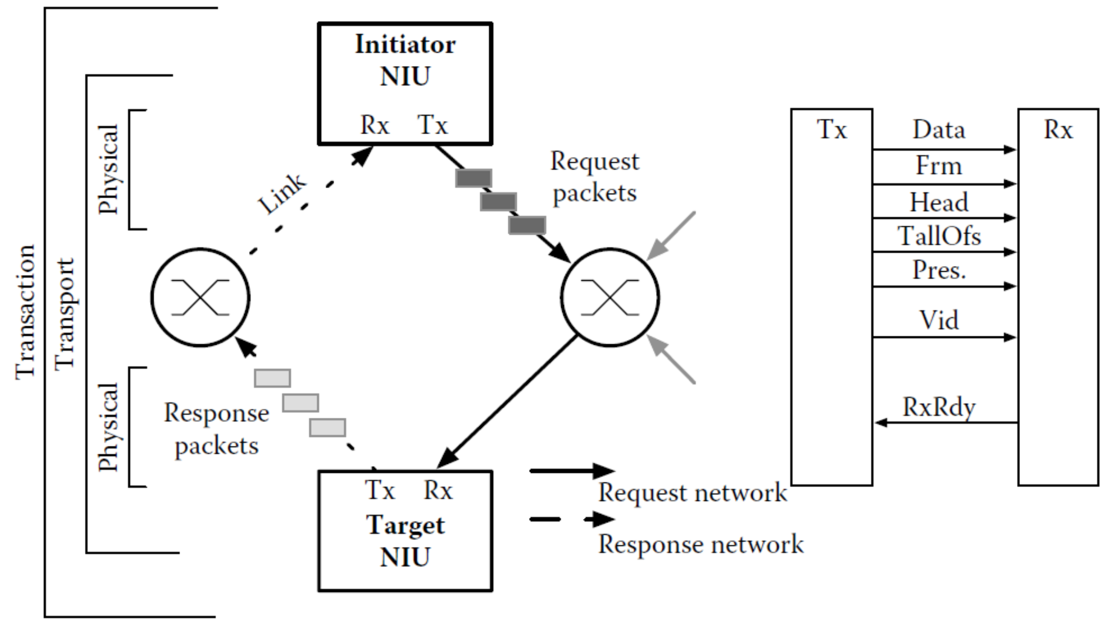
<table>
<tr><th>'800 Patent Claim</th><th>Motorola Product Including Snapdragon System on Chip [1]</th></tr>
<tr><td></td><td>


FIGURE 11.1
NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI.

See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313.

The Target NIUs are "used to connect a slave node to the NoC":
</td></tr>
</table>

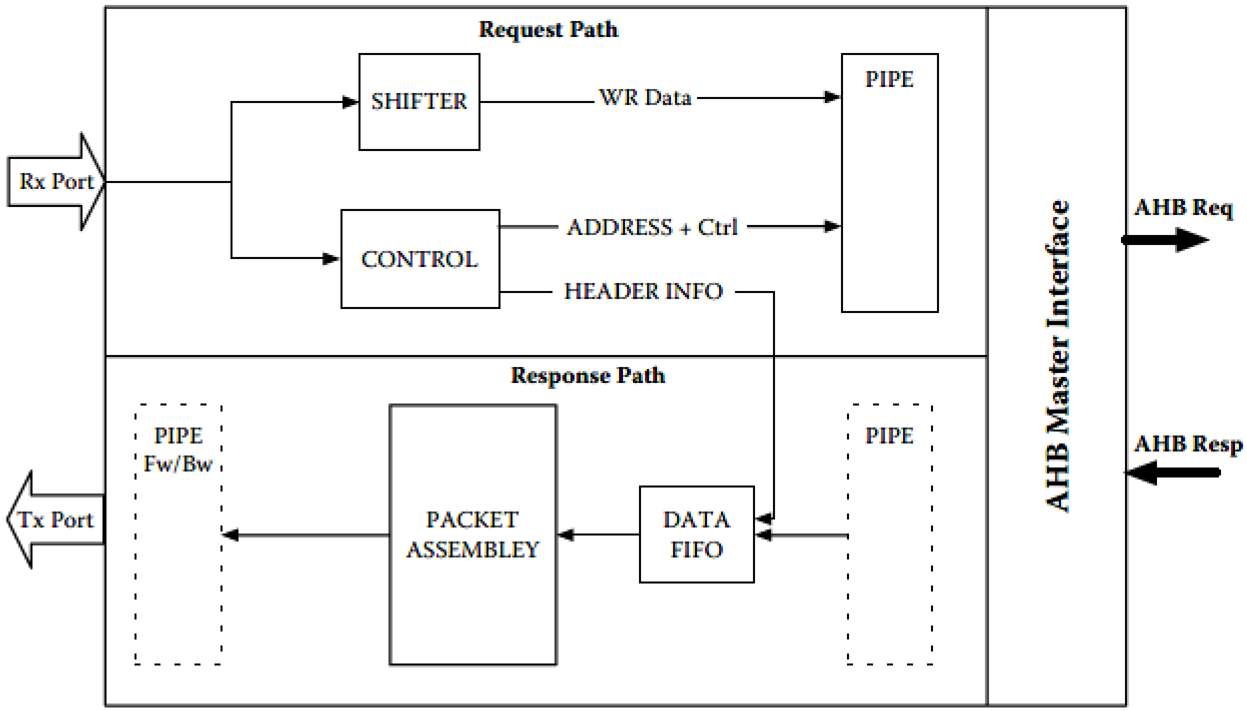
'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets" and includes blocks such as "Data FIFO "and "Packet Assembly":

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	11.3.2.2 Target NIU Units Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Target NIU Architecture  **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	In the Target NIUs of the Arteris NoC, similar to as described above for the Initiator NIUs, "[a] FIFO memory is inserted in the datapath for AHB … accesses. The FIFO memory absorbs data at the AHB … rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is defined by the hardware parameter" which "indicates the amount of data required to generate a … packet: each time the FIFO is full, a … packet is sent on the Tx port": A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

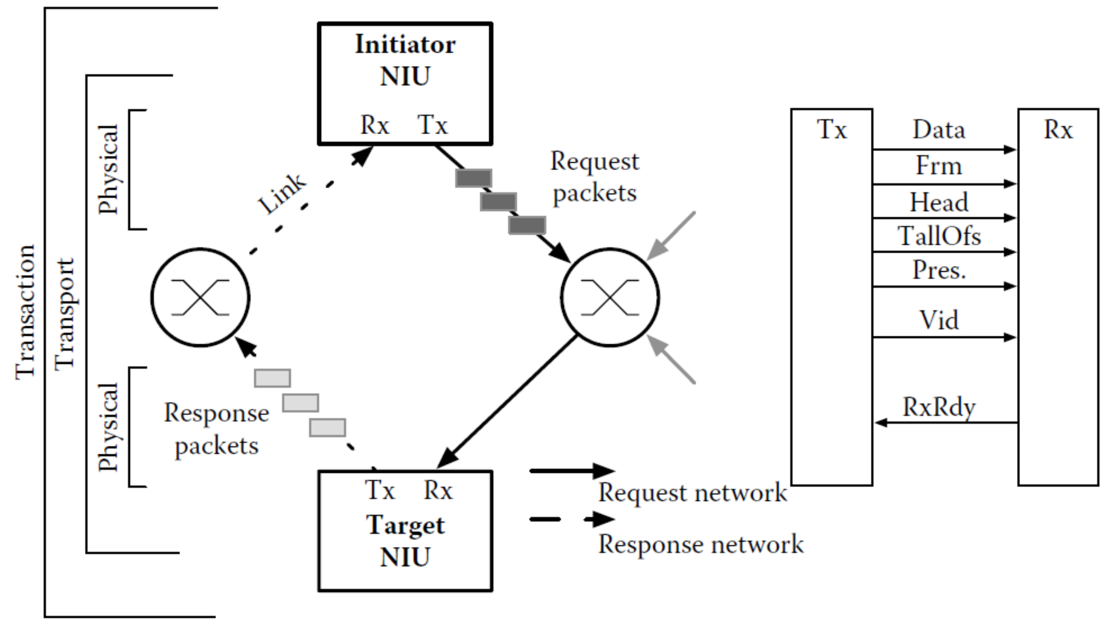
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.
buffering by the slave wrapper of the slave interface unit data from the slave to be transferred over the interconnect until a first optimal amount of data is buffered;	The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product buffers by the slave wrapper of the slave interface unit data from the slave to be transferred over the interconnect until a first optimal amount of data is buffered, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network: ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

4856-4191-2641.3

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Target NIUs are "used to connect a slave node to the NoC":

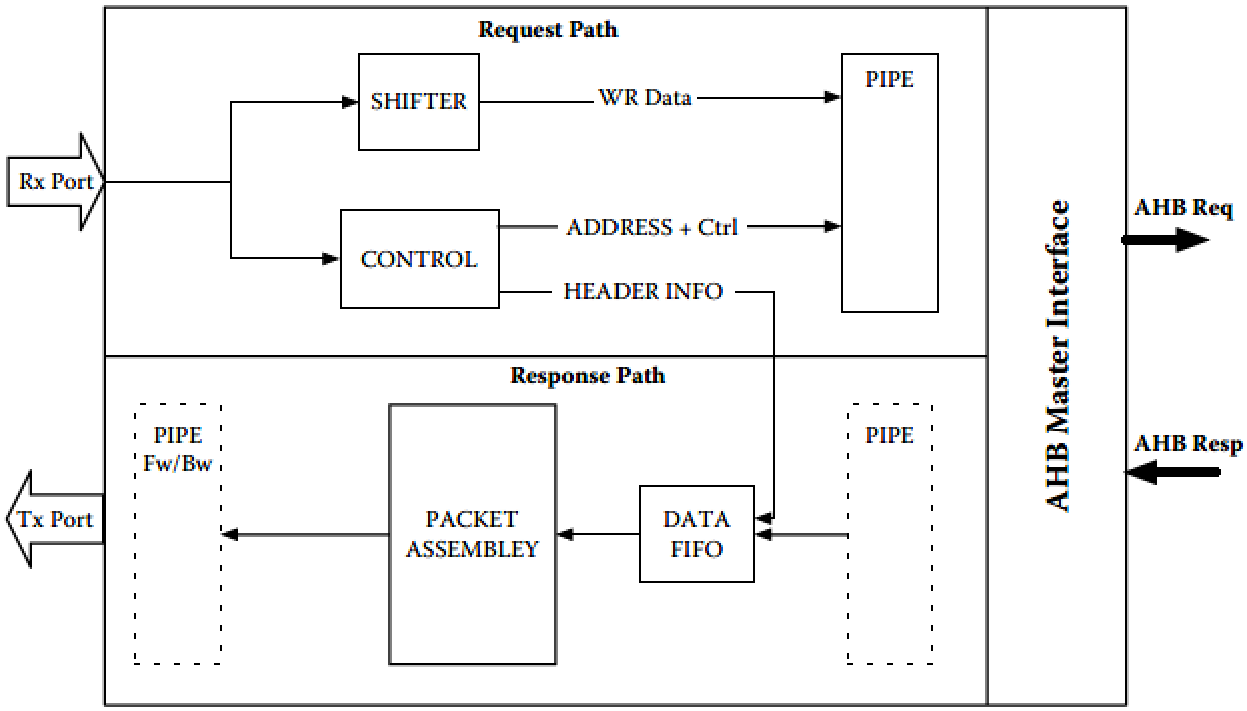
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets" and includes blocks such as "Data FIFO "and "Packet Assembly":

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	11.3.2.2 Target NIU Units Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

4856-4191-2641.3

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Target NIU Architecture  **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	In the Target NIUs of the Arteris NoC, similar to as described above for the Initiator NIUs, "[a] FIFO memory is inserted in the datapath for AHB … accesses. The FIFO memory absorbs data at the AHB … rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is defined by the hardware parameter" which "indicates the amount of data required to generate a … packet: each time the FIFO is full, a … packet is sent on the Tx port": A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318. As a further illustration, the Arteris NoC uses "a mechanism called rated adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency." For other traffic, the "[b]est effort traffic can be left untouched[,]" "[l]atency sensitive traffic may have its urgency modulated as a function of the transaction[,]" "[s]oft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives[,]" and "[o]n the real-time modem data port, the hurry is fixed at a critical level": Those effects can be mended by the insertion of buffering. In the case of peak bandwidth reduction, a simple FIFO does the job: Busy states present at the output of the FIFO do not propagate back to the input until the FIFO is full. For a peak bandwidth increase, the situation is a bit more complex. In a FIFO, wait states present at the input are only absorbed when the FIFO is not empty. Arteris proposes a mechanism called rate adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency. In this second step, the architecture is modified to introduce some buffering. In our example 760 bytes of memory have been distributed across the topology. Some have been put on existing links; some required the creation of new links. See Application driven network-on-chip architecture exploration & refinement for a complex SoC, https://www.arteris.com/hs-fs/hub/48858/file-14363521-pdf/docs/springerappdrivennocarchitecture8.5x11.pdf, at pg.16.
transferring the buffered data from the slave wrapper to the master	The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product transfers the buffered data from the slave wrapper to the master wrapper when said first optimal amount of data has been buffered by the slave wrapper, either literally or under the doctrine of equivalents.

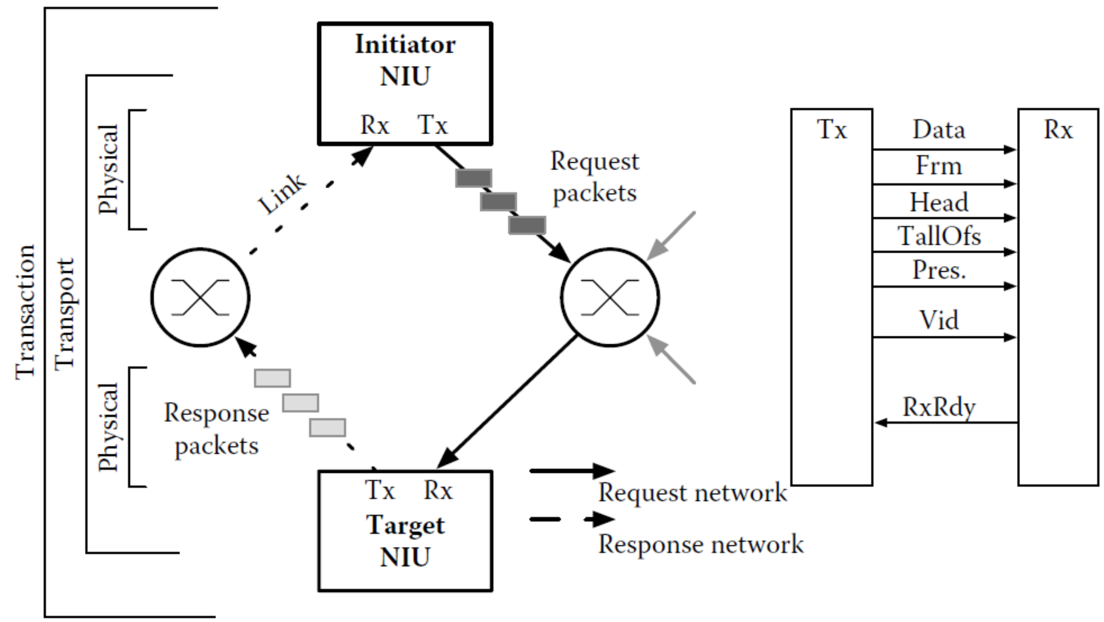
4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
wrapper when said first optimal amount of data has been buffered by the slave wrapper;	For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network: ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

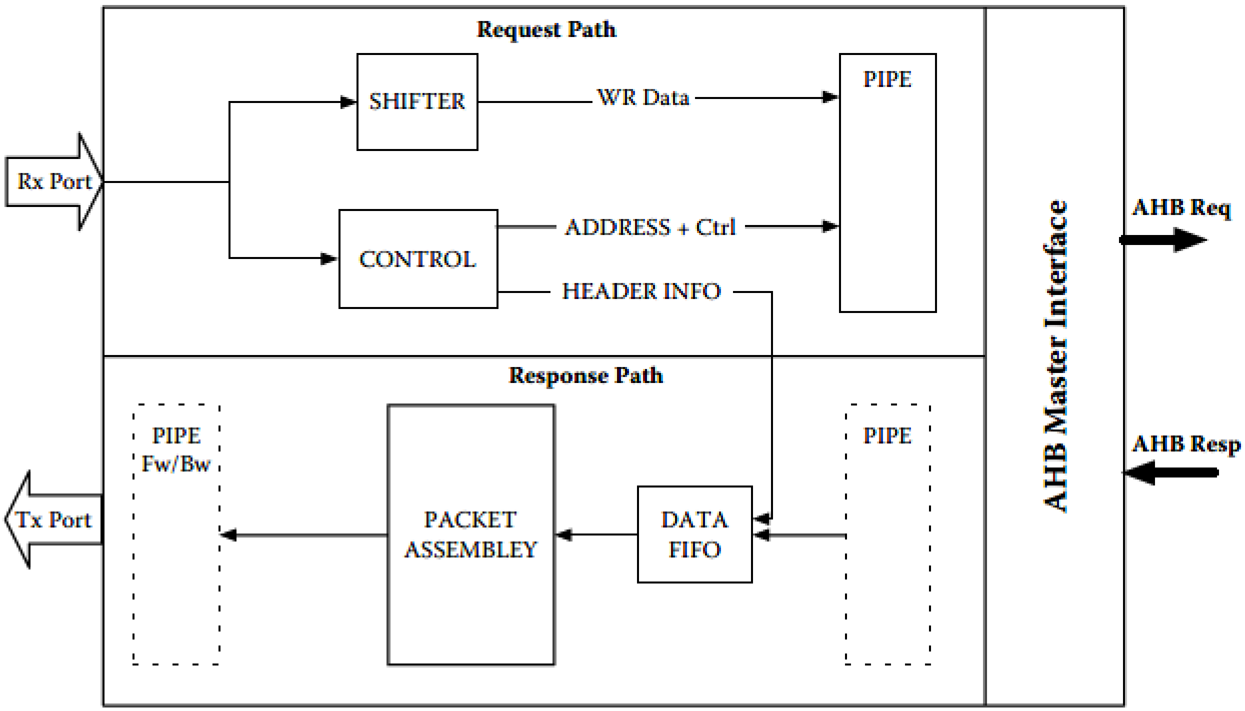
'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Target NIUs are "used to connect a slave node to the NoC":

4856-4191-2641.3

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets" and includes blocks such as "Data FIFO "and "Packet Assembly":

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	11.3.2.2 Target NIU Units Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Target NIU Architecture  **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	In the Target NIUs of the Arteris NoC, similar to as described above for the Initiator NIUs, "[a] FIFO memory is inserted in the datapath for AHB … accesses. The FIFO memory absorbs data at the AHB … rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is defined by the hardware parameter" which "indicates the amount of data required to generate a … packet: each time the FIFO is full, a … packet is sent on the Tx port": A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

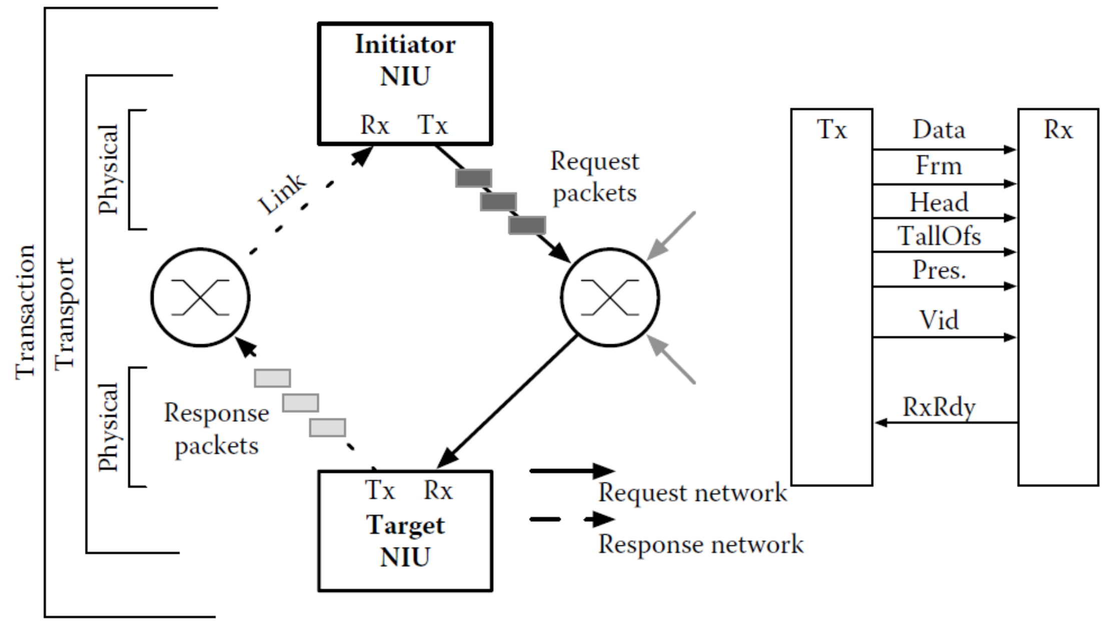
'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.
buffering by the master wrapper of the master interface unit data from the master to be transferred over the interconnect until a second optimal amount of data is buffered by the master wrapper;	The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product buffers by the master wrapper of the master interface unit data from the master to be transferred over the interconnect until a second optimal amount of data is buffered by the master wrapper, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network: ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

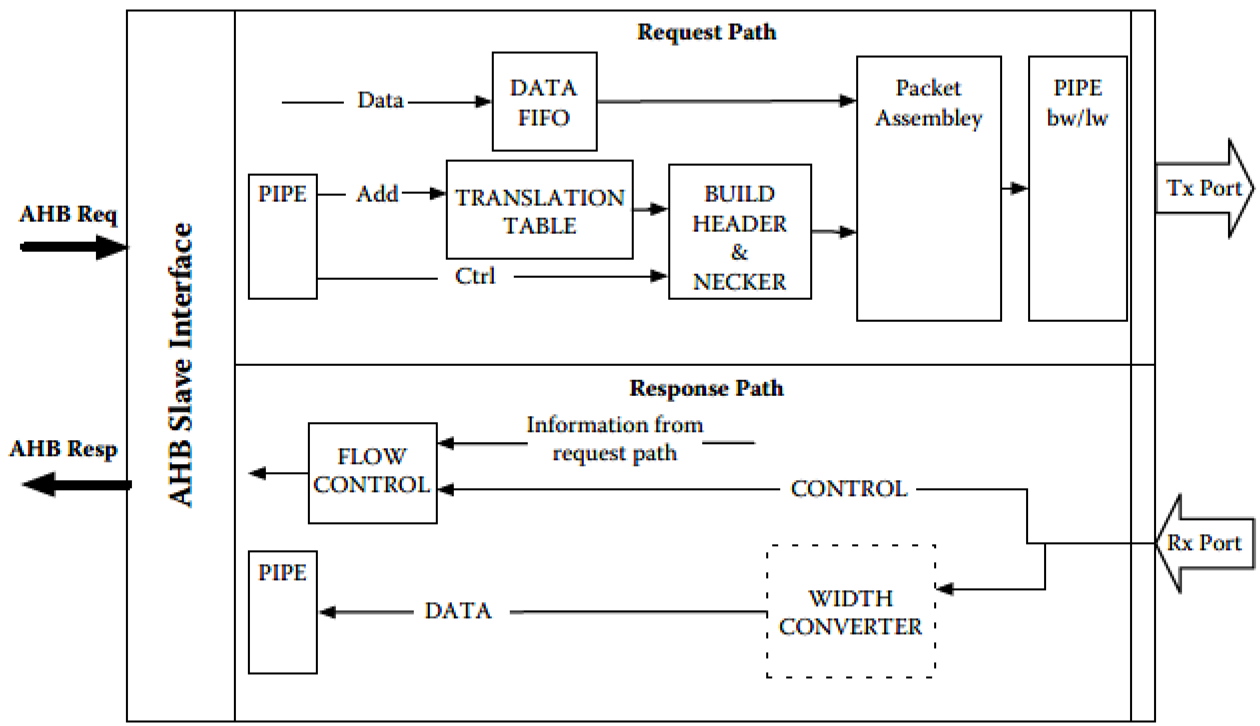
'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC":

4856-4191-2641.3

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. In the Arteris NoC "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC" and includes blocks such as "Data FIFO," "Translation Table," "Build Header & Necker," and "Packet Assembly":

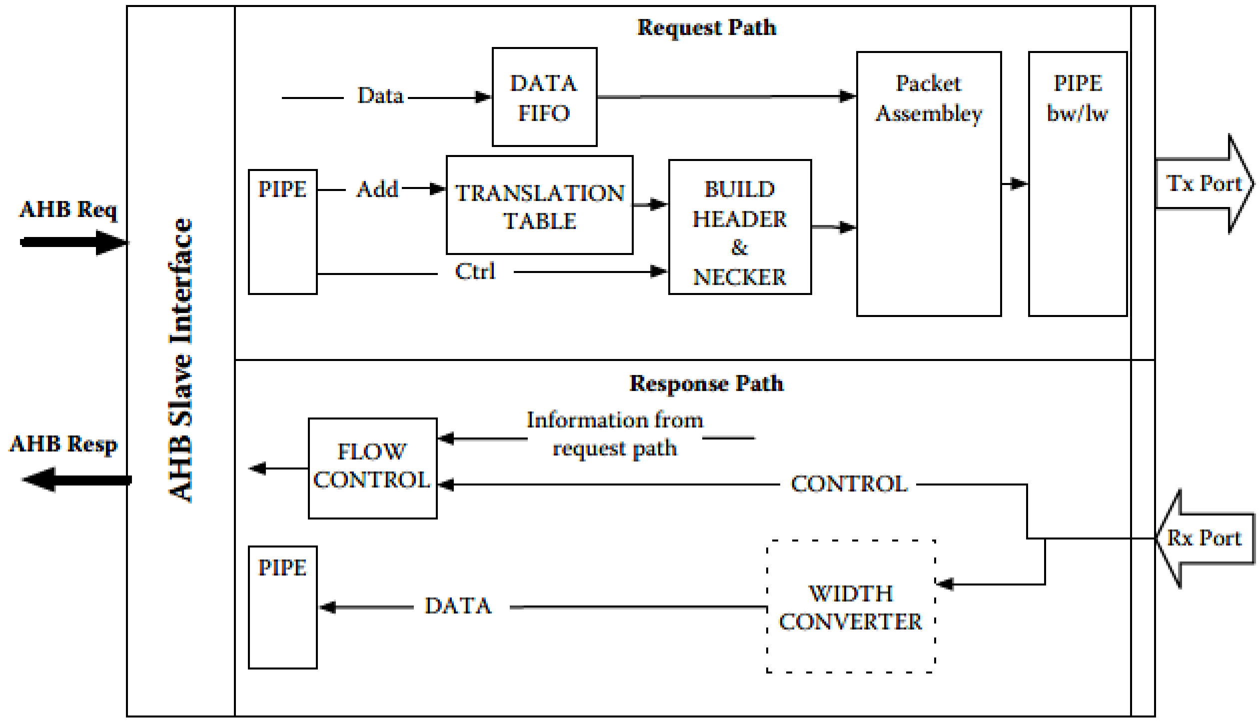
'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	NIU Architecture  **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317. In the Initiator NIUs of the Arteris NoC, a "FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	defined by the hardware parameter" which "indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port": ***11.3.2.1 Initiator NIU Units*** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	NIU Architecture  **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318. As a further illustration, the Arteris NoC uses "a mechanism called rated adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency." For other traffic, the "[b]est effort traffic can be left untouched[,]" "[l]atency sensitive traffic may have its

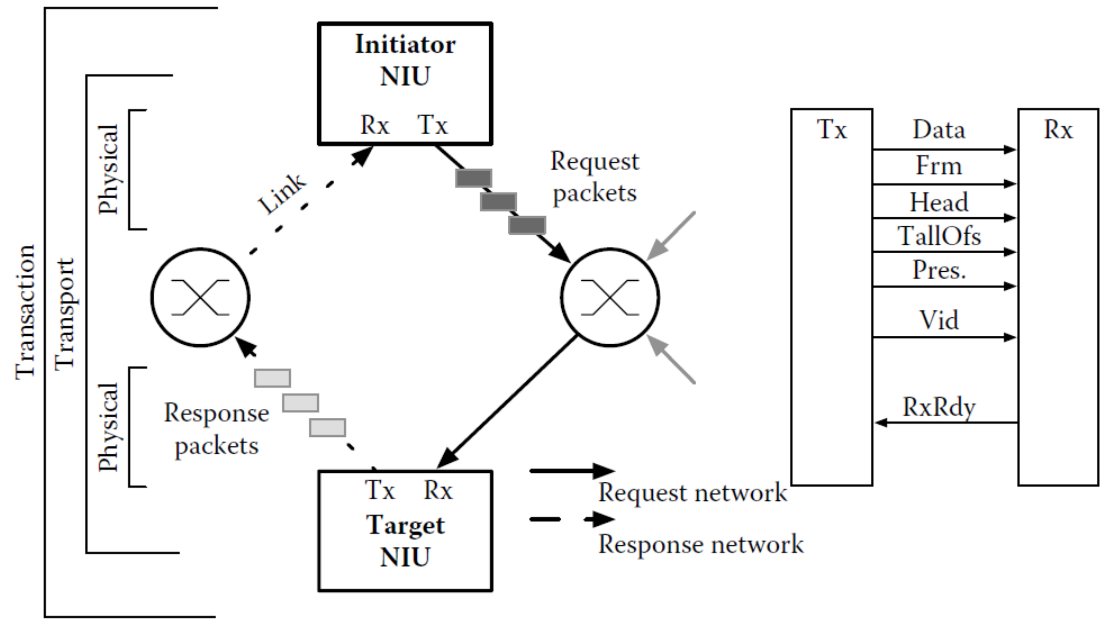
4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	urgency modulated as a function of the transaction[,]" "[s]oft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives[,]" and "[o]n the real-time modem data port, the hurry is fixed at a critical level": Those effects can be mended by the insertion of buffering. In the case of peak bandwidth reduction, a simple FIFO does the job: Busy states present at the output of the FIFO do not propagate back to the input until the FIFO is full. For a peak bandwidth increase, the situation is a bit more complex. In a FIFO, wait states present at the input are only absorbed when the FIFO is not empty. Arteris proposes a mechanism called rate adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency. In this second step, the architecture is modified to introduce some buffering. In our example 760 bytes of memory have been distributed across the topology. Some have been put on existing links; some required the creation of new links. See Application driven network-on-chip architecture exploration & refinement for a complex SoC, https://www.arteris.com/hs-fs/hub/48858/file-14363521-pdf/docs/springerappdrivennocarchitecture8.5x11.pdf, at pg.16.
transferring the buffered data from the master wrapper to the slave wrapper when said second optimal amount of data has been buffered by the master wrapper,	The Arteris NoC utilized by the Snapdragon SoC included in the Motorola product transfers the buffered data from the master wrapper to the slave wrapper when said second optimal amount of data has been buffered by the master wrapper, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network:

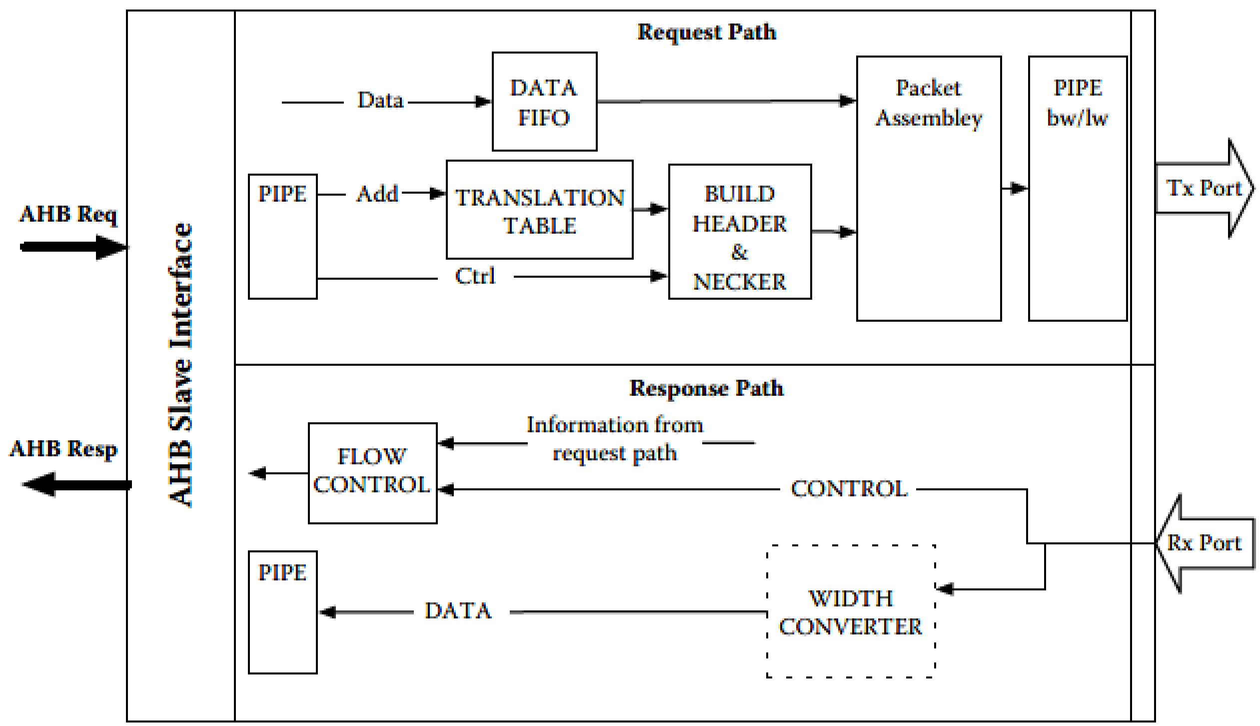
'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.1.1 Transaction Layer The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC":

4856-4191-2641.3

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. In the Arteris NoC "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC" and includes blocks such as "Data FIFO," "Translation Table," "Build Header & Necker," and "Packet Assembly":

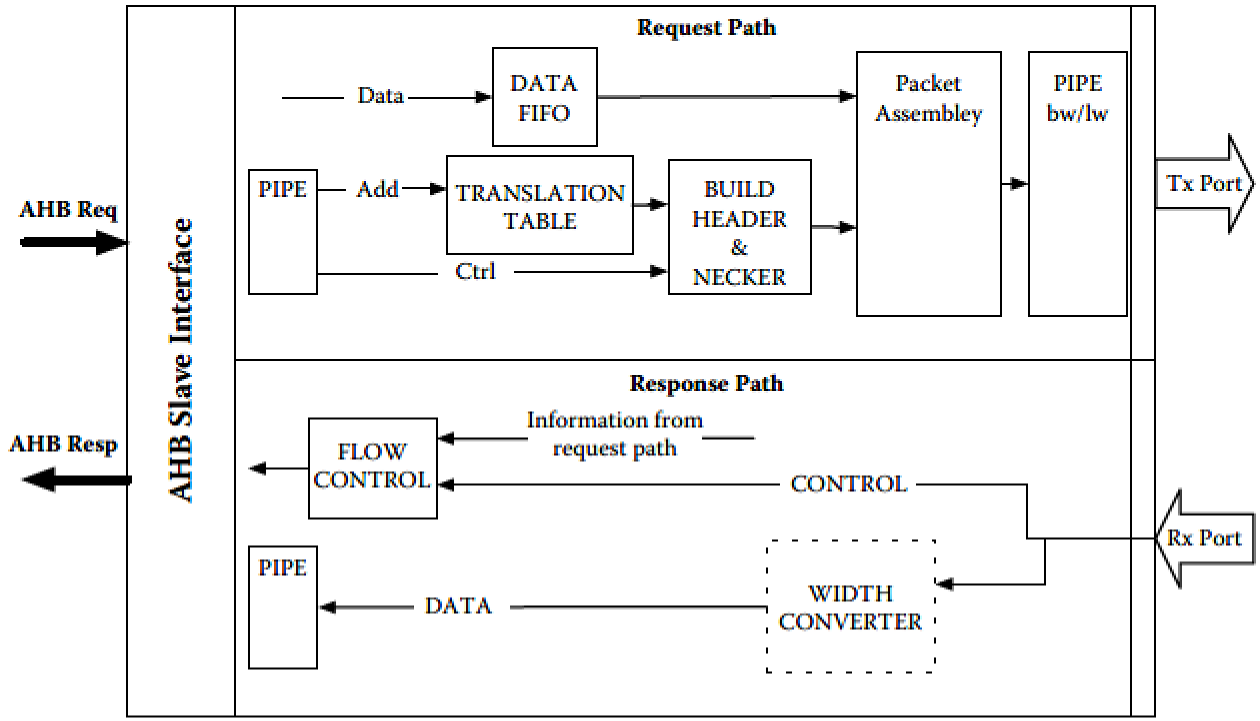
'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	NIU Architecture  **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317. In the Initiator NIUs of the Arteris NoC, a "FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	defined by the hardware parameter" which "indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port": ***11.3.2.1 Initiator NIU Units*** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	NIU Architecture  **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318.
wherein at least one of the first determination unit	In the Arteris NoC utilized by the Snapdragon SoC included in the Motorola product, at least one of the first determination unit and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
and the second determination unit is further configured to determine an optimal moment for sending the data in said first wrapper or said second wrapper according to communication properties of the communication between the master and the slave, wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it	according to communication properties of the communication between the master and the slave wherein the communication properties include ordering of data transport, flow control including when a remote buffer is reserved for a connection, then a data producer will be allowed to send data only when it is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery, either literally or under the doctrine of equivalents. For example, the Arteris NoC uses Network Interface Units (NIUs), which "translate[] between third-party [OCP, AMBA AHB, APB, and AXI protocols] and NTTP protocols" and in the Arteris NoC, the NIUs "are at the boundary of the NoC" and there is a NIU connected to each of the master and slave nodes, between the nodes and the network: ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

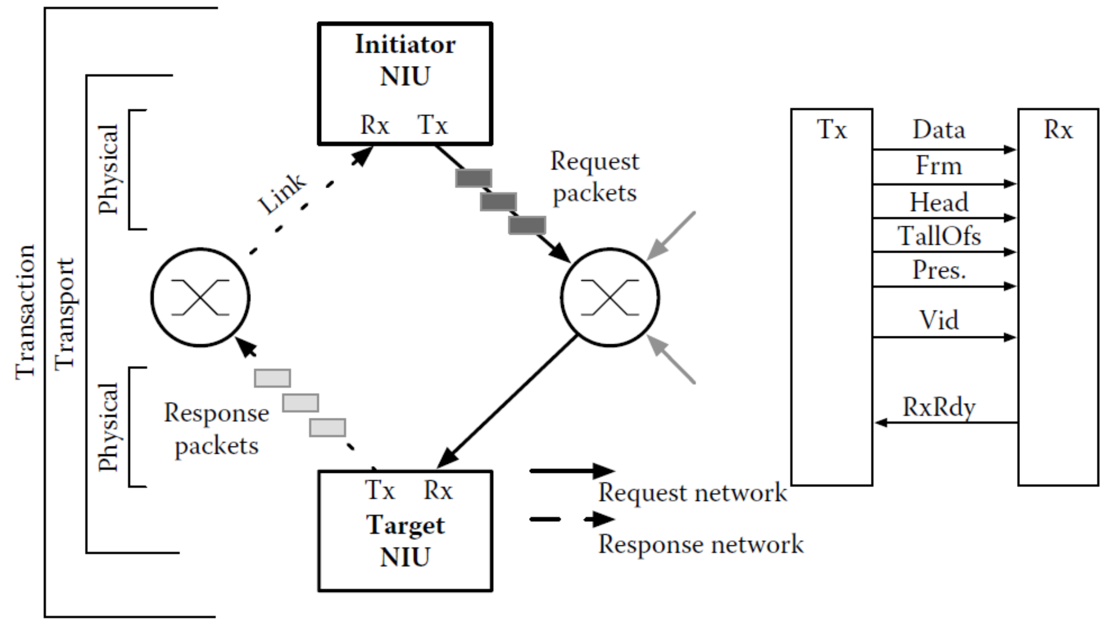
4856-4191-2641.3

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
is guaranteed that space is available for the produced data at the remote buffer, throughput where a lower bound on throughput is guaranteed, latency where an upper bound for latency is guaranteed, lossiness including dropping of data, transmission termination, transaction completion, data correctness, priority, and data delivery.	on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

4856-4191-2641.3

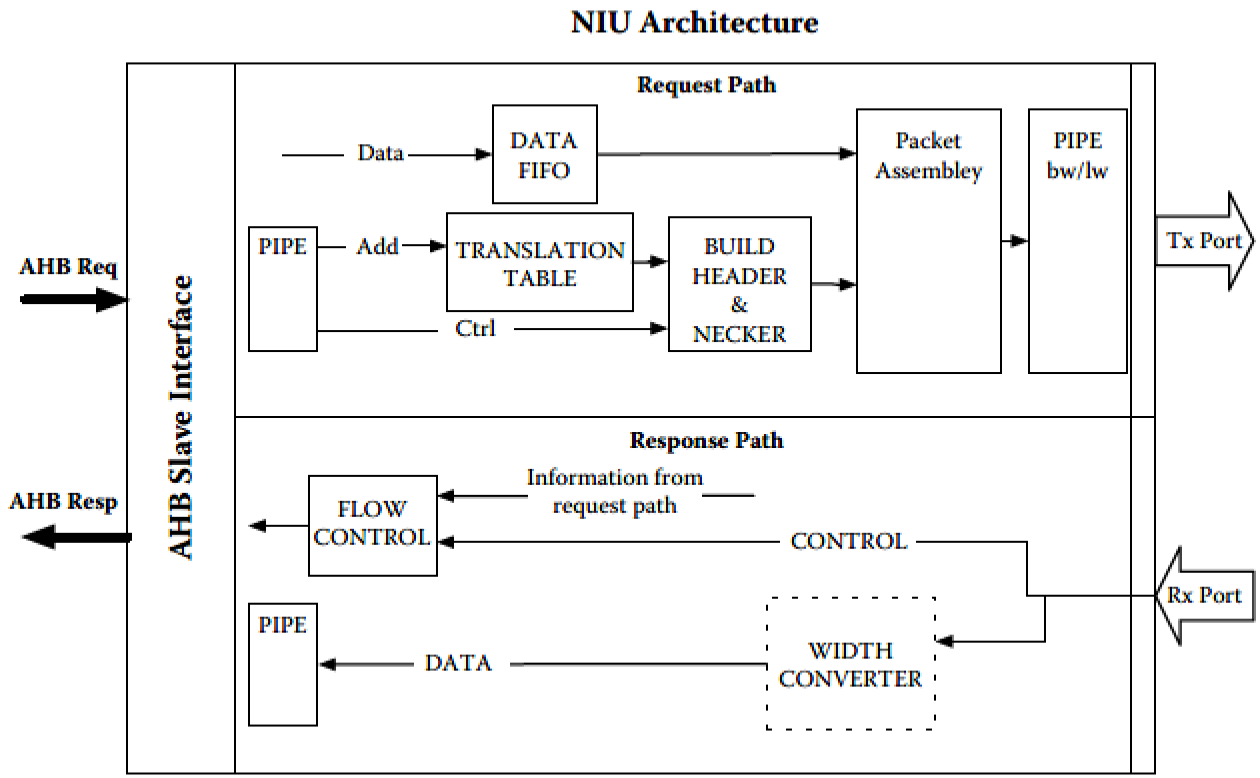
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 311, 312-313. The Initiator NIUs are "used to connect a master node to the NoC" and the Target NIUs are "used to connect a slave node to the NoC":

4856-4191-2641.3

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2 Network Interface Units The Arteris Danube IP library includes NIUs for different third party protocols. Currently, three different protocols are supported: AHB (APB), OCP, and AXI. For each protocol, two different NIU units can be instantiated: • **Initiator NIU**—third party protocol-to-NTTP, used to connect a master node to the NoC • **Target NIUs**—NTTP-to-third party protocol, used to connect a slave node to the NoC Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 316-317. In the Arteris NoC "Initiator NIU units…enable connection between an AMBA-AHB master IP and the NoC" and includes blocks such as "Data FIFO," "Translation Table," "Build Header & Necker," and "Packet Assembly":

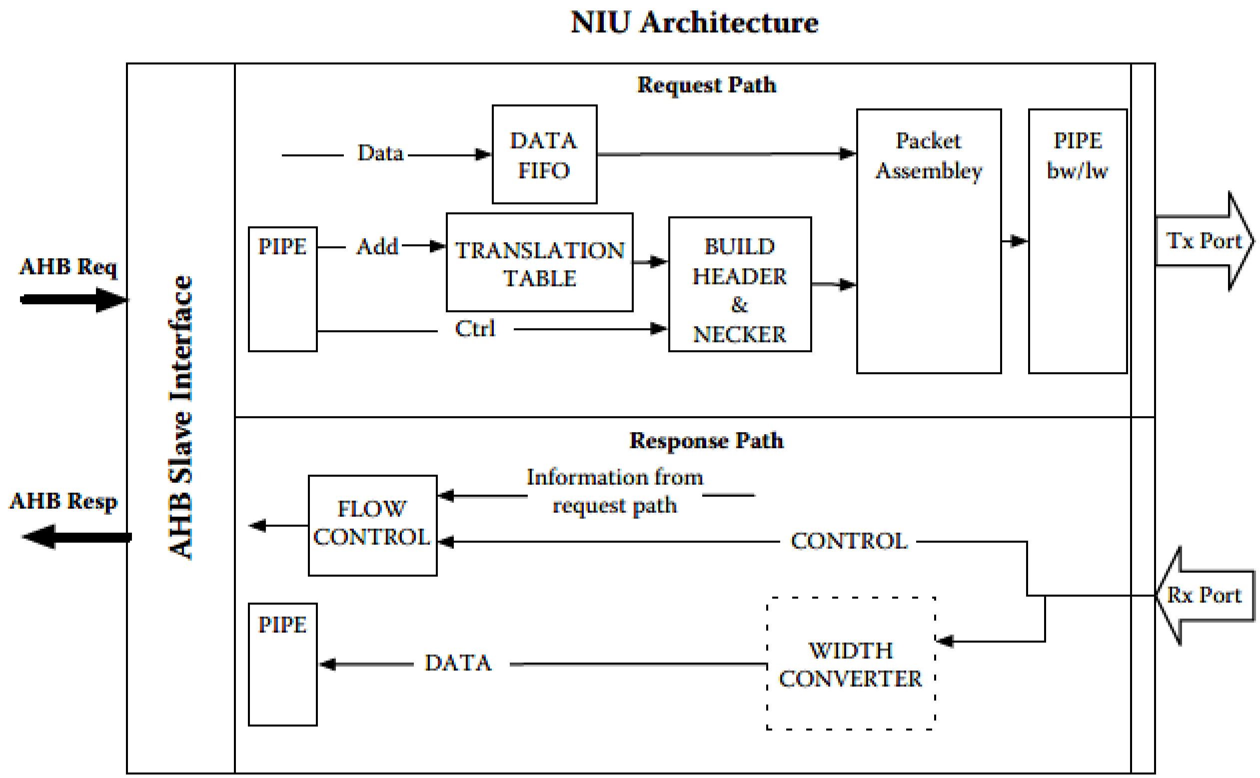
U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317. In the Initiator NIUs of the Arteris NoC, a "FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	defined by the hardware parameter" which "indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port": ***11.3.2.1 Initiator NIU Units*** Initiator NIU units (the architecture of the AHB initiator is given in Figure 11.4) enable connection between an AMBA-AHB master IP and the NoC. It translates AHB transactions into an equivalent NTTP packet sequence, and transports requests and responses to and from a target NIU, that is, slave IP (slave can be any of the supported protocols). The AHB-to-NTTP unit instantiates a Translation Table for address decoding. This table receives 32-bit AHB addresses from the NIU and returns the packet header and necker information that is needed to access the NTTP address space: Slave address, Slave offset, Start offset, and the coherency size (see Figure 11.2). Whenever the AHB address does not fit the predefined decoding range, the table asserts an error signal that sets the error bit of the corresponding NTTP request packet, for further error handling by the NoC. The translation table is fully user-defined at design time: it must first be completed with its own hardware parameters, then passed to the NIU. A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can

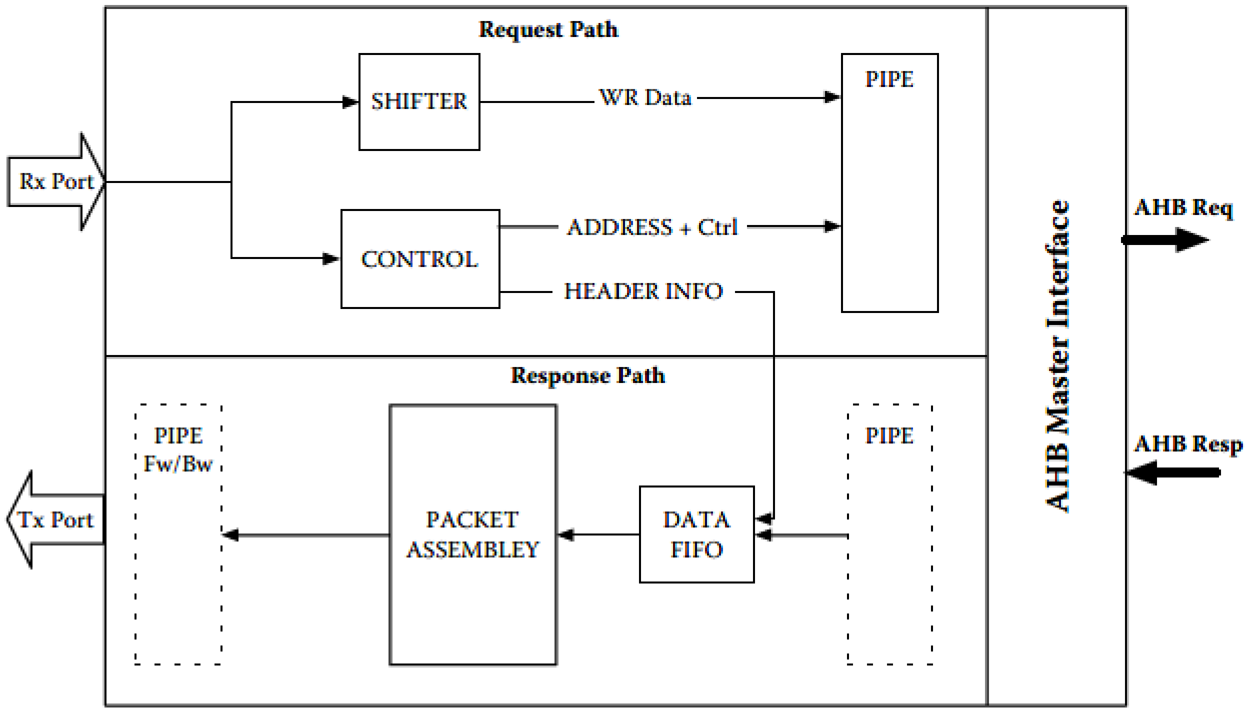
4856-4191-2641.3

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	 **FIGURE 11.4** Network interface unit: Initiator architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318. As further example, "Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets" and includes blocks such as "Data FIFO "and "Packet Assembly":

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	11.3.2.2 Target NIU Units Target NIU units enable connection of a slave IP to the NoC by translating NTTP packet sequences into equivalent packet transactions, and transporting requests and responses to and from targets (the architecture of the AHB Target NIU is given in Figure 11.5). For the AHB target NIU, the AHB address space is mapped from the NTTP address space using the slave offset, the start/stop offset, and the slave address fields, when applicable (from the header of the request packet, Figure 11.2). The AHB address bus is always 32 bits wide, but the actual address space size may be downsized by setting a hardware parameter. Unused AHB address bits are then driven to zero. The NTTP request packet is then translated into one or more corresponding AHB accesses, depending on the transaction type (word aligned or nonaligned access). For example, if the request is an atomic Store, or a Load that can fit an AHB burst of specified length, then such a burst is generated. Otherwise, an AHB burst with unspecified length is generated.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Target NIU Architecture  **FIGURE 11.5** Network interface unit: Target architecture. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 318-319.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	In the Target NIUs of the Arteris NoC, similar to as described above for the Initiator NIUs, "[a] FIFO memory is inserted in the datapath for AHB … accesses. The FIFO memory absorbs data at the AHB … rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received." "[T]he FIFO depth is defined by the hardware parameter" which "indicates the amount of data required to generate a … packet: each time the FIFO is full, a … packet is sent on the Tx port": A FIFO memory is inserted in the datapath for AHB write accesses. The FIFO memory absorbs data at the AHB initiator rate, so that NTTP packets can burst at NoC rate as soon as a minimum amount of data has been received. The width of the FIFO and the AHB data bus is identical, and the FIFO depth is defined by the hardware parameter. This parameter indicates the amount of data required to generate a Store packet: each time the FIFO is full, a Request packet is sent on the Tx port. Of course, if the AHB access ends before the FIFO is full, the NTTP request packet is sent. Because AHB can only tolerate a single outstanding transaction, the AHB bus is frozen until the NTTP transaction has been completed. That is • During a read request, until the requested data arrives from the Rx port • During a nonbufferable write request, in which case only the last access is frozen and the acknowledge occurs when the last NTTP response packet has been received • When an internal FIFO is full

4856-4191-2641.3

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 317-318. As a further illustration, the "Arteris NTTP protocol is packet-based" and the packets, which have "header and necker cells [that] contain information relative to routing, payload size, packet type, and the packet target address," are "transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes": ***11.3.1.2 Transport Layer*** The Arteris NTTP protocol is packet-based. Packets created by NIUs are transported to other parts of the NoC to accomplish the transactions that are required by foreign IP nodes. All packets are comprised of cells: a header cell, an optional necker cell, and possibly one or more data cells (for packet definition see Figure 11.2; further descriptions of the packet can be found in the next subsection). The header and necker cells contain information relative to routing, payload size, packet type, and the packet target address. Formats for request packets and response packets are slightly different, with the key difference being the presence of an additional cell, the necker, in the request packet to provide detailed addressing information to the target. *Id.* at 313. As yet a further illustration, packets in the Arteris NoC are "delivered as words that are sent along links and "[o]ne link (represented in Figure 11.1) defines the following signals," which include "the current priority of the packet used to define preferred traffic class (or Quality of Service)" and "[f]low control":

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	• **Data**—Data word of the width specified at design-time. • **Frm**—When asserted high, indicates that a packet is being transmitted. • **Head**—When asserted high, indicates the current word contains a packet header. When the link-width is smaller than single (SGL), the header transmission is split into several word transfers. However, the Head signal is asserted during the first transfer only. • **TailOfs**—Packet tail: when asserted high, indicates that the current word contains the last packet cell. When the link-width is smaller than single (SGL), the last cell transmission is split into several word transfers. However, the Tail signal is asserted during the first transfer only. • **Pres.**—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2). • **Vld**—Data valid: when asserted high, indicates that a word is being transmitted. • **RxRdy**—Flow control: when asserted high, the receiver is ready to accept word. When de-asserted, the receiver is busy. *Id.* at 313-314.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	As a further illustration, the Arteris NoC implements Quality of Service (QoS) to "provide[] a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic"; "QoS, which includes guarantees of throughput and/or latency, is achieved by exploiting the signal pressure embedded into the NTTP packet definition" where the "pressure signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed"; and the "pressure information will be embedded in the NTTP packet at the NIU level": **Quality of Service (QoS).** The QoS is a very important feature in the interconnect infrastructures because it provides a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic. Usually the end users are looking for guarantees on bandwidth and/or end-to-end communication latency. Different mechanisms and strategies have been proposed in the literature. For instance, in Æthereal NoC [11,24] proposed by NXP, a TDMA approach allows the specification of two traffic categories [25]: BE and GT. In the Arteris NoC, the QoS is achieved by exploiting the signal pressure embedded into the NTTP packet definition (Figures 11.1 and 11.2). The pressure

4856-4191-2641.3

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed. For example, we can imagine associating the generation of the pressure signal when a certain threshold has been reached in the FIFO of the corresponding IP. This pressure information will be embedded in the NTTP packet at the NIU level: packets that have pressure bits equal to zero will be considered without QoS; packets with a nonzero value of the pressure bit will indicate preferred traffic class.* Such a QoS mechanism offers immediate service to the most urgent inputs and variables, and fair service whenever there are multiple contending inputs of equal urgency (BE). Within switches, arbitration decisions favor preferred packets and allocate remaining bandwidth (after preferred packets are served) fairly to contending packets. When there are contending preferred packets at the same pressure level, arbitration decisions among them are also fair. The Arteris NoC supports the following four different traffic classes:

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	• **Real time and low latency (RTLL)**—Traffic flows that require the lowest possible latency. Sometimes it is acceptable to have brief intervals of longer latency as long as the average latency is low. Care must be taken to avoid starving other traffic flows as a side effect of pursuing low latency. • **Guaranteed throughput (GT)**—Traffic flows that must maintain their throughput over a relatively long time interval. The actual bandwidth needed can be highly variable even over long intervals. Dynamic pressure is employed for this traffic class. • **Guaranteed bandwidth (GBW)**—Traffic flows that require a guaranteed amount of bandwidth over a relatively long time interval. Over short periods, the network may lag or lead in providing this bandwidth. Bandwidth meters may be inserted onto links in the NoC to regulate these flows, using either of the two methods. If the flow is assigned high pressure, the meter asserts backpressure (flow control) to prevent the flow from exceeding a maximum bandwidth. Alternatively, the meter can modulate the flows pressure (priority) dynamically as needed to maintain an average bandwidth. • **Best effort (BE)**—Traffic flows that do not require guaranteed latency or throughput but have an expectation of fairness.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	* Note that in the NTTP packet, the pressure field allows more then one bit, resulting in multiple levels of preferred traffic. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 315-316. In addition, the Arteris Interconnect includes "a mechanism called rated adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency." For other traffic, the "[b]est effort traffic can be left untouched[,]" "[l]atency sensitive traffic may have its urgency modulated as a function of the transaction[,]" "[s]oft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives[,]" and "[o]n the real-time modem data port, the hurry is fixed at a critical level." Those effects can be mended by the insertion of buffering. In the case of peak bandwidth reduction, a simple FIFO does the job: Busy states present at the output of the FIFO do not propagate back to the input until the FIFO is full. For a peak bandwidth increase, the situation is a bit more complex. In a FIFO, wait states present at the input are only absorbed when the FIFO is not empty. Arteris proposes a mechanism called rate adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency. In this second step, the architecture is modified to introduce some buffering. In our example 760 bytes of memory have been distributed across the topology. Some have been put on existing links; some required the creation of new links. Application driven network-on-chip architecture exploration & refinement for a complex SoC, https://www.arteris.com/hs-fs/hub/48858/file-14363521-pdf/docs/springer-appdrivennocarchitecture8.5x11.pdf, at p. 16.

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	For the other traffic, the configuration can be done in architecture. • Best effort traffic can be left untouched. • Latency sensitive traffic may have its urgency modulated as a function of the transaction: *Normal* for writes and *important* for reads. • Soft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives: *Critical* until a specified bandwidth is obtained on a sliding 4 microsecond window, and *normal* thereafter. These settings are set through configuration registers and may be modified while the interconnect is running. The mechanism is called a bandwidth regulator. • On the real-time modem data port, the hurry is fixed at a critical level. *Id.* at 18. As a further illustration, the Arteris NoC implements QoS mechanisms that performs arbitration based on "Bandwidth Regulartor (BR)" and "Bandwidth Limiter (BL)":

U.S. Patent No. 8,086,800 (Radulescu and Goossens)
"Integrated circuit and method for buffering to optimize burst length in networks on chips"

'800 Patent Claim	Motorola Product Including Snapdragon System on Chip [1]
	Bandwidth Limiters and Rate Regulators Many times architects will want to implement QoS within their SoC but the QoS prioritization data is not available from the individual IP blocks. In this case, QoS information may be generated from within the NoC interconnect using Arteris' QoS Generator. The QoS Generator can instantiate sophisticated, and software programmable, means to regulate interconnect QoS, including: › Bandwidth Limiters – Bandwidth limiters cause a socket to stop accepting requests when a run-time programmable throughput threshold has been exceeded. › Rate Regulators – Rate regulators cause a socket's transactions to be demoted when a bandwidth threshold is reached. This can be considered a smoother version of the bandwidth limiter because transactions are only demoted instead of stalled. https://www.arteris.com/end-to-end-quality-of-service-qos